Reconocimiento y clasificación de escenas utilizando descriptores semánticos.

Andrés Mauricio Aguirre Montoya

Universidad Tecnológica de Pereira

Facultad de ingenierías

Programa de ingeniería eléctrica

Pereira, 9 de abril de2021

Reconocimiento y clasificación de escenas utilizando descriptores semánticos.

Andrés Mauricio Aguirre Montoya

Trabajo de grado presentado como requisito para optar al título de

Ingeniero Electricista

Director:

PhD. Luis Hernando Ríos González

Universidad Tecnológica de Pereira

Facultad de ingenierías

Programa de ingeniería eléctrica

Pereira, 9 de abril de2021

Nota de aceptación

_____

_____

_____

_____

Luis Hernando Ríos González

Pereira, 9 de abril de 2021

Contenido

Agradecimientos

Principalmente a Dios por permitirme vivir esta experiencia y la obtención de una meta, a mi padre Víctor Fabio Aguirre Atehortúa por su acompañamiento y apoyo, a mi madre Luz Enid Montoya Londoño por su paciencia y consejos durante este periodo de formación, a mi hermana Lizeth Dayana Aguirre Montoya por la compañía y apoyo durante el largo periodo de estudio y al resto de mi familia en general.

A la Universidad Tecnológica de Pereira y al programa de ingeniería eléctrica por permitirme hacer parte del proceso, a los docentes que acompañaron y guiaron esta etapa de formación académica en cada una de sus asignaturas, al director de este proyecto de grado el ingeniero Luis Hernando Ríos González por sus aportes, guía y dedicación en la realización de este documento.

Resumen

Con la evolución en el campo de visión por computador, se encuentran constantemente propuestas de mejoras a los métodos tradicionales de clasificación y representación de imágenes. Desde propuestas basadas en la explotación de características por medio de redes neuronales artificiales hasta métodos de representación de imágenes basados en extracción de características locales. Dado que los algoritmos de clasificación de imágenes aumentan su eficacia en función de características aprendidas previamente y los algoritmos de representación de imágenes por características locales tratan de explotar rasgos distintivos en muestras de estudio disminuyendo el efecto de variaciones como recortes o giros en la capacidad de clasificar eficientemente, el objetivo de este trabajo será el análisis de resultados de clasificación según el método elegido para extraer características y discriminar imágenes distintas entre sí que pueden o no tener ciertos rasgos comunes.

Para llevarlo a cabo se evalúa tanto el efecto de usar arquitecturas de redes neuronales entrenadas previamente con miles de imágenes similares, como las variaciones en los métodos basados en características locales al momento de agrupar y codificar la información extraída de las imágenes. En este proceso analizar la relación entre los resultados de clasificación en metodologías basadas en CNN(convolutional neural networks), VLAD (Vector Locally Aggregated descriptor) o BOVW (Bag of visual words) puede generar indicios sobre las características que mejor están en capacidad de explotar cada una de las metodologías presentadas y el tipo de imágenes para las cuales son más eficientes.

Abstract

With the evolution in the field of computer vision, there are constantly proposals for improvements to the traditional methods of classification and representation of images. From proposals based on the exploitation of characteristics by means of artificial neural networks to image representation methods based on extraction of local characteristics. Since image classification algorithms increase their efficiency based on previously learned characteristics and image representation algorithms by local characteristics try to exploit distinctive features in study samples by reducing the effect of variations such as clipping or twists in the ability to classify efficiently, the objective of this work will be the analysis of classification results according to the chosen method to extract characteristics and discriminate different images from each other that may or may not have certain common features.

To do this, we evaluate both the effect of using previously trained neural network architectures with thousands of similar images, as well as the variations in methods based on local characteristics when grouping and encoding the information extracted from the images. In this process, analyzing the relationship between the classification results in methodologies based on CNN (convolutional neural networks), VLAD (Vector Locally Aggregated descriptor) or BOVW (Bag of visual words) can generate indications about the characteristics that best They are able to exploit each of the methodologies presented and the type of images for which they are most efficient.

Índice de tablas

Índice de figuras

Capítulo 1. Introducción

Como camino para lograr solucionar problemas de clasificación en diversos tipos de datos, se ha recurrido a diversas herramientas del aprendizaje automático como lo son : redes neuronales artificiales, máquinas de soporte vectorial, arboles aleatorios, entre otros. En el caso de la clasificación de imágenes es usual recurrir a la primera estructura mencionada, las redes neuronales artificiales especialmente las redes que usan las operaciones de convolución y agrupación como base para aprender características de las imágenes de entrada. Comparten en cierta forma la propiedad de propagar información entre neuronas ajustando parámetros internos de cada capa o grupo de neuronas para minimizar el error en la salida. Tomando como base la capacidad del cerebro humano para reconocer patrones en las imágenes que percibe como, por ejemplo, reconocer el rostro de una persona a partir de los rasgos que lo definen un modelo de red neuronal artificial podría aprender a reconocer, generalizar y discriminar patrones complejos a partir de la exposición de este modelo a un gran número de imágenes de entrada. Así como las redes neuronales basadas en la cantidad de parámetros a ajustar y la profundidad en término del número de capas que contiene puede detectar características desde muy genéricas hasta específicas, los algoritmos basados en extracción de características locales pueden resultar una alternativa válida a los problemas de clasificación de imágenes. Usando diferentes estrategias es posible detectar puntos claves de una imagen que pueden determinar el contenido de esta, como la detección de esquinas, cambios de contraste o bordes que entregan información relevante de los datos de entrada. Las estadísticas calculadas para cada imagen de entrenamiento son susceptibles a agrupar y posteriormente discriminar por medio de un proceso de codificación y posterior clasificación[21][22][25]

Con el fin de analizar el alcance de cada método de clasificación propuesto, se seleccionan un conjunto de imágenes de entrada divididas en entrenamiento y validación. Cada subconjunto denominado base de imágenes puede contener ciertas diferencias entre las etapas de entrenamiento y validación, por ejemplo, entrenar un modelo capaz de reconocer escenas y validar su funcionamiento con objetos o partes de la escena original. Este grupo de imágenes además de ser evaluado en distintas combinaciones de escenas y objetos como la anterior, estarán inmersos tanto en distintas arquitecturas de redes neuronales artificiales como VGG16 y Resnet50 como en distintos algoritmos de extracción de características y codificación como pueden ser VLAD, Fisher Vector o BOVW. Los resultados obtenidos de la aplicación de los anteriores métodos a cada base de imágenes serán consignados tanto en tablas de precisión vs

algoritmo como en graficas que permitan la observación de los modelos más eficientes que han sido entrenados.

1.1 Definición del problema

Los algoritmos de reconocimiento de escenas y objetos han tenido un gran impacto en el campo de la visión por computador. Basados en el uso de redes neuronales artificiales han permitido clasificar y detectar imágenes de gran complejidad en diversos campos de la ciencia. Si bien las redes neuronales convolucionales han entregado buenos resultados en las etapas de extracción de características y clasificación estas por sí solas son sensibles a cambios en las escenas a evaluar como la presencia de objetos no clasificables o la dificultad de diferenciar los límites entre dos objetos de interés, dando resultados muy generales, omitiendo puntos importantes o confundiendo entre distintas categorías. [4][5][6]

Existen en la actualidad diversas estructuras de extracción de características y representación de escenas agrupadas en dos tipos de análisis : análisis artesanales (análisis a mano) que buscan representar las imágenes a partir de características retinotopicas, extraídas directamente de las capas convolucionales de la red; y por otro lado existen análisis basados en probabilidades de ocurrencia los cuales representan la escena como un conjunto de probabilidades al interior de un modelo estadístico que reduce el efecto negativo de los objetos comunes y las variaciones entre escenas. En resumen, el problema radica en representar las escenas de forma tal que la red aprenda a clasificar y reconocer objetos de forma precisa, pero aprovechando las ventajas potenciales de las capas convolucionales. Esto se logra capturando las características que diferencian cada clase con mayor independencia del entorno en el cual se ubican los objetos. [4][5][6]

Aun cuando los métodos tradicionales basados en mapas de características directamente extraídas de la red pueden funcionar de forma adecuada con ciertas imágenes, el complemento con técnicas estadísticas que modelen la ocurrencia de objetos puede adaptar mejor el algoritmo de reconocimiento a distintos problemas y facilitar el uso de técnicas de transferencia de aprendizaje para reutilizar complejas y robustas estructuras desarrolladas anteriormente. Por lo tanto, la pregunta de investigación se formula de la siguiente manera: ¿ Como explotar las características discriminantes de los objetos en una escena y que metodología se debe aplicar si se desea hacer uso de redes de clasificación pre-entrenadas?

1.2 Justificación del problema

El problema de la representación de escenas nace de la necesidad de explotar las capacidades propias de redes neuronales entrenadas previamente en tareas de clasificación y detección. Se busca mejorar los resultados de clasificación local en los distintos parches o sectores de la escena , de forma que se pueda reducir el error de clasificación de los algoritmos, así como adaptarlos a variaciones en las imágenes y presencia de objetos comunes. Consiste en modelar una imagen como una distribución estadística de probabilidades de ocurrencia de los objetos. A diferencia de los algoritmos de clasificación directos que solo usan vectores de probabilidad posterior de toda la imagen, el algoritmo de clasificación local busca clasificar las imágenes a partir de parches que contienen los objetos de interés en dentro de la imagen. Al mapear una imagen tomando los vectores de probabilidad posterior de sus objetos se están aprovechando las características extraídas a través de las distintas capas de la red, lo que implica obtener una representación aún más precisa, teniendo de entrada una tasa de precisión garantizada.[4][5]

Tomando como objetivo fundamental el desarrollo de un algoritmo de reconocimiento que explote de forma más profunda las características que definen un objeto en el interior una escena, es posible adaptarse a imágenes de mayor complejidad en cuanto a clasificación y reconocimiento se refiere. De ahí la importancia de describir una metodología que permita desde la etapa de preparación y segmentación de imágenes hasta la detección de objetos y reconocimiento de la escena evaluada que pueda además ser compatible con la representación de imagen propuesta. Para el caso particular, la prueba de algoritmos de reconocimiento de escenas se plantea como un reto para imágenes tipo interiores, debido a la composición de este tipo de escenas en las cuales se puede ver una múltiple cantidad de objetos que varían notablemente de posición entre una imagen y otra, además de presentar objetos que se repiten escena tras escena sin ser objetos de clasificación. En una imagen así se hace necesaria la extracción y representación local de las mismas con el fin de determinar de forma más precisa la relación entre el objeto y el medio, información que las capas convolucionales por sí mismas aun no alcanzan a extraer.[4][5]

Para la presentación de el algoritmo de detección y reconocimiento propuesto se hace énfasis en la preparación de imágenes de entrenamiento y validación de la red neuronal de forma tal que permita a la red neuronal convolucional clasificar previamente cada objeto de una forma más precisa debido a un proceso de etiquetado que busca demarcar la forma física de los objetos y evitar la clasificación errónea de objetos que no son de interés en la etapa de detección y reconocimiento. Así como se busca en la etapa de clasificación local la selección e implementación de una estructura que se adapte a la base de

entrenamiento propia pero además contenga la complejidad y robustez de una red pre-entrenada en este tipo de imágenes. En cuanto a las etapas de extracción de características y representación de imágenes el punto clave se encuentra en aprovechar resultados de clasificación con un nivel previo de acierto que contienen la información capturada en los procesos de convolución de la red neuronal. Así a partir de modelos estadísticos usados con frecuencia en tareas relacionadas con visión por computador, el algoritmo usa los descriptores semánticos de Fisher como una abstracción de toda esta información, para mapearla y llevarla a un espacio matemático en el cual sea posible realizar una predicción más precisa, computacionalmente más eficiente y con menor sensibilidad a la composición de la escena. [4][5]

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un algoritmo de reconocimiento de escenas que permita la explotación local de sus características y una representación de la imagen que contrarreste los efectos negativos de objetos comunes y variaciones entre escenas.

1.3.2 Objetivos específicos

a. Revisar y estudiar la bibliografía especializada sobre tareas de clasificación, detección y reconocimiento de objetos orientadas al uso de vectores de Fisher.
b. Analizar y seleccionar desde la información previamente obtenida los modelos de redes neuronales artificiales más adecuados para el problema de reconocimiento de escenas de interiores, así como la metodología más adecuada para procesar las características extraídas de las imágenes.
c. Preparar la base de imágenes que serán usadas en la etapa de entrenamiento de cada algoritmo. Además, haciendo uso de herramientas de etiquetado para definir las categorías a tratar, se presentan las distintas posibilidades de procesamiento como segmentación o detección de objetos.
d. Diseñar un algoritmo que permita agrupar las tareas de representación y clasificación de imágenes.
e. Evaluar el algoritmo diseñado con un grupo de imágenes de prueba y analizar los resultados respecto a los métodos globales de representación y clasificación .

1.4 Estado del arte

En esta sección se presenta una serie de propuestas que aplican la extracción local de características como herramienta para la posterior representación y clasificación de escenas.

1.4.1 Reconocimiento de escenas con objetividad

El método de reconocimiento de escenas con objetividad es un método de explotación local de características basado en la probabilidad de ocurrencia de un objeto en una escena. Busca representar una escena más allá de mapas retinotopicos obtenidos por medio de capas convolucionales, es decir utiliza vectores de probabilidad posterior para agrupar los parches representativos de las escenas como distribuciones estadísticas, de esta forma captura la información discriminatoria de los objetos y filtra los parches que no contienen información relevante. Obtenidos los conjuntos de probabilidades de coocurrencia se generan descriptores semánticos locales que en conjunto forman la representación global de la escena. Con una imagen representada por objetos discriminantes los vectores de probabilidad finalmente obtenidos ingresan en una estructura de clasificación de escenas capaz de diferenciar que tipo de escena fue ingresada a la red neuronal. [4][7]

La metodología de reconocimiento de escenas descrita anteriormente hace énfasis en dos partes puntuales del proceso: representación y clasificación de escenas. Debido a que sobre estas dos tareas se refleja gran parte de la tasa de precisión del sistema. [4]

1.4.2 Representación de escenas

En la representación de escenas existen dos técnicas de extracción de características : representación manual basada en la explotación de características holísticas y locales extraída directamente desde la información de texturas en la imagen, generando descriptores de alto nivel, pero sensibles a las variaciones en las imágenes y computacionalmente costosos de generar. Por otro lado, están los descriptores de imagen basados en aprendizaje los cuales explotan las características a través de capas convolucionales y capas completamente conectadas de la red neuronal. Existen varias implementaciones las cuales varían el grupo de capas desde el cual se extrae la información. En estas últimas implementaciones basadas en aprendizaje se pretende extraer las características de las escenas con independencia entre clases ignorando los efectos negativos de los objetos comunes. [4][7][9]

1.4.3 Clasificación de escenas

En la tarea de clasificación de escenas se han propuesto múltiples modelos que buscan describir y clasificar la escena a partir de las características extraídas localmente en la fase de representación. Los modelos implementados se agrupan principalmente en dos categorías : modelos generativos y modelos discriminativos. Los modelos generativos usados en este tipo de tareas por lo general usan técnicas basadas en bayesiano jerárquico para obtener la relación entre la escena a clasificar y la información de los objetos que la componen .[15][14]

Al usar los objetos como descriptores de la escena puede clasificar la imagen en una categoría aun con relaciones muy diversas como ocurre por ejemplo en escenas de interiores. Algunos clasificadores generativos muy usados incluyen modelos ocultos de Márkov (HMM, campos aleatorios (MRF) y asignación de Dirichlet latente (LDA). [4][17][16]

1.4.4 Clasificación de escenas con vectores semánticos de Fisher

Esta metodología de clasificación parte del uso de redes neuronales convolucionales (CNN por sus siglas en ingles) para clasificar la escena por parches de imagen. Usando las capas completamente conectadas de la red y la capa de salida softmax se obtienen vectores de probabilidad que de forma similar a la anterior propuesta pretende modelar la escena como una bolsa de probabilidades locales llamada bolsa de semántica (BoS por sus siglas en ingles), es así como a partir de parches locales discriminativos se termina llevando la información a una serie de vectores de probabilidad que en conjunto se denomina imagen BoS y es una fiel representación de la presencia de objetos de interés en la escena. [13][10] Debido a la complejidad de clasificar la imagen en este espacio de la imagen BoS se usa una incrustación semántica de Fisher la cual mapea este conjunto previamente obtenido y explota las características más representativas de la escena en función de los resultados de clasificación local de la red neuronal convolucional. El vector de Fisher es así una abstracción de la imagen BoS que obtienen información desde un espacio no euclidiano hasta un espacio euclidiano en el cual se han obtenido mejores resultados tanto en exactitud a la hora de clasificar las escenas como eficiencia computacional en el instante de calcular este tipo de descriptores.[5][18]

Las investigaciones se han centrado en dos tipos de implementaciones : la primera pretende modelar la imagen BoS como una mezcla de Dirichlet con el fin de obtener un vector gradiente de Fisher que haga la tarea de mapeo y descripción de características. La segunda implementación tiene como objetivo utilizar descriptores semánticos como parámetros de una distribución multinomial. En esta segunda

implementación el vector semántico de Fisher se calcula como una mezcla gaussiana resultante del modelado estadístico de la imagen BoS. El obtenerse a partir de un conjunto de vectores de probabilidad contenidos en la imagen BoS lo hace un complemento ideal en tareas de detección de objetos así como clasificación y reconocimiento de escenas debido a la posibilidad de explotar los beneficios potenciales de las redes neuronales convolucionales y al mismo tiempo generar representaciones de imágenes más exactas que ataquen los efectos negativos de objetos comunes y variaciones entre escenas que han sido demarcados como los principales problemas a solucionar. [5]

1.4.5 Aplicación de vectores semánticos de Fisher en la clasificación de escenas de alta resolución

Esta propuesta de clasificación de imágenes de gran resolución aparece como una herramienta para el procesamiento de imágenes obtenidas por teledetección, lo cual consiste en utilizar como imágenes de entrada para una red neuronal convolucional capturas satelitales que son recibidas de forma remota. Las redes neuronales convolucionales han mostrado buenos resultados en el procesamiento de imágenes por teledetección debido a su funcionamiento mismo, puede aprender características jerárquicas que permiten describir el contenido de una imagen.[11][2][3][12]

Con el potencial explorado anteriormente en CNN y haciendo énfasis especial en VGG-Net (Visual Geometry Group Network) Se propone la utilización de las capas convolucionales como extractores de características fusionando tanto la representación global a través de las capas completamente conectadas como el método de vector mejorado de Fisher para obtener un vocabulario o codificación. Para el caso del vector mejorado de Fisher se hace necesario interpretar la quinta capa convolucional conformada por conv5.1, conv5.2 y conv5.3 como descriptores de características, es decir el proceso de creación de un vocabulario busca relacionar las características convolucionales extraídas en estas capas con la escena a la cual pertenecen. De forma resumida, el proceso se completa en tres fases : extracción de características por CNN, creación de vocabulario a partir de las características de la capa quinta capa convolucional y finalmente la aplicación de estrategias para reducir dimensionalidad del vector de características. Una vez se logra la fusión de los descriptores extraídos en las capas completamente conectadas y el vector codificado de Fisher desde las capas convolucionales es posible utilizar como método de clasificación una máquina de soporte vectorial. Las representaciones obtenidas se hacen basadas en pesos pre-entrenados en Imagenet Large Scale Visual Recognition Competition (ILSVRC), esto debido a que los sistemas de teledetección no generan las cantidades necesarias de datos para entrenar un nuevo modelo de CNN. [1]

1.5 Alcance del trabajo de grado

A través de este documento se pretende presentar de forma simplificada y precisa dos metodologías distintas para la representación y clasificación de escenas. Tomando como punto de partida la información contenida tanto en el estado del arte como en el marco teórico, se describe el proceso de entrenamiento y validación de dos arquitecturas de redes neuronales artificiales, más precisamente VGG-16 y Resnet50 pre entrenadas en grandes bases de imágenes como lo son Imagenet y Places365. En cuanto al soporte teórico de las mismas, si bien no se entrara en detalle en cuanto al complejo modelo matemático generado por medio de la propagación de información entre capas y actualización de gradientes, es posible observar desde definiciones muy generales los principios de funcionamiento presentes en redes neuronales convolucionales. En cuanto a la representación de imágenes por medio de características locales se definen las principales etapas que componen el proceso, como lo son : extracción de características, generación de descriptores, codificación y representación de escenas tanto para los métodos basados en mezclas gausianas como los que soportan su funcionamiento en la agrupación por K-Means.

Aunque las metodologías presentadas están limitadas en este caso a tareas de clasificación, se menciona y presenta de forma breve, la aplicación de herramientas de etiquetado de imágenes a las bases propias de entrenamiento a manera de ilustración tanto para problemas de segmentación de imágenes como problemas de detección de objetos.

1.6 Estructura del trabajo de grado

Este documento está compuesto por 5 capítulos distribuidos de la siguiente forma : El capítulo 1 será un capítulo de introducción, aquí se encontrara una descripción corta del trabajo, el planteamiento y justificación del problema a tratar, el estado del arte, el objetivo general y los objetivos específicos que ayudaran a cumplirlo. En el capítulo 2 esta toda la fundamentación teórica que va desde la clasificación de los problemas de aprendizaje automático hasta las definiciones de cada etapa en el proceso de clasificación por redes neuronales y la codificación de vectores de Fisher basados en puntos de interés. El capítulo 3 presenta la metodología a implementar, en la cual se describe de forma secuencial como se prepararon los datos de entrenamiento para su posterior procesamiento, las estrategias para extraer características tanto por CNN como por vectores de Fisher hasta las variantes realizadas en cada propuesta en la búsqueda de mejores resultados. En el capítulo 4 se presentan los resultados obtenidos para cada caso planteado en la metodología sobre los distintos grupos de imágenes a utilizar, aquí se compara el rendimiento de los algoritmos basados en redes neuronales convolucionales frente a los métodos de extracción de características locales en función de la cantidad de aciertos y errores. Finalmente, en el capítulo 5 se presentan las conclusiones del presente trabajo obtenidas en relación con los resultados del capítulo 4.

Capítulo 2. Marco teórico

En el proceso de clasificar imágenes por medio de algoritmos de inteligencia artificial como pueden ser redes neuronales profundas, maquinas de soporte vectorial, métodos de clasificación por bosques aleatorios entre otros intervienen muchos factores desde el preprocesamiento de la imagen hasta la toma de decisiones y estimaciones de probabilidad que en ultima son los resultados de clasificación. En este capitulo se abordaran conceptos básicos en el entrenamiento y validación de una red neuronal profunda, así como definiciones asociadas a las herramientas que se utilizaran. Finalmente se mencionaran algunos métodos de extracción de características locales y representación de imágenes a partir de algoritmos de agrupación o clustering muy usado en aprendizaje no supervisado. [25].

2.1 Aprendizaje automático

El aprendizaje automático es un campo de investigación conocido por unir el conocimiento de distintas ramas. Por ejemplo, en algunos libros lo ubican en medio de la informática, la estadística y las matemáticas debido a los conceptos necesarios para explicar la forma en que funcionan sus algoritmos. La funcion de sus algoritmos es aprender a reconocer características de grandes conjuntos de datos de forma que los identifique y al recibir información nueva pueda relacionarla con las características aprendidas en el pasado, su funcion es intentar predecir a partir de la experiencia.

Según la forma en que recibe y explota características de los datos se pueden clasificar en : aprendizaje supervisado, aprendizaje no supervisado, aprendizaje por refuerzo. El aprendizaje supervisado intentara relacionar características de un conjunto de datos con la etiqueta que trae este por defecto, por ejemplo, para clasificar perros y gatos por medio de una red neuronal profunda hay que enseñarle a la red que rasgos y detalles definen al perro por medio de un gran conjunto de fotos de perros. En el aprendizaje no supervisado el algoritmo debe aprender de forma automática a encontrar las características de cada conjunto de datos y agruparla de tal forma que se pueda hacer a una representación de él. Su funcionamiento aumenta en complejidad porque este no tiene etiquetas que relacionan la información que entra con una clase, debe aprender a discriminar información según la forma que está compuesta. En el aprendizaje por refuerzo se busca a partir de bonificaciones y penalizaciones guiar al algoritmo hacia un punto en donde el error sea pequeño, un ejemplo claro son los sistemas de inteligencia artificial de muchos juegos en los que el algoritmo ha tenido que explorar caminos y penalizar o bonificar según el acierto de las decisiones así cada programa de este tipo aprende a reaccionar a información nueva.[21][22]

2.2 Redes neuronales profundas

Una red neuronal profunda se puede referir al conjunto de redes neuronales que tienen una mayor cantidad de grupos con neuronas conectados entre si . En ciertas palabras es un tipo de algoritmo para aprendizaje automático en el cual un conjunto de capas de representación está conectado de forma sucesiva desde la entrada de la red hasta la etapa de clasificación y salida de la red neuronal artificial. Su termino profundo hace referencia al numero de capas de representación que puede llegar a tener este tipo de estructuras desde decenas hasta cientos. Cada capa es un conjunto de neuronas artificiales que reciben información de las capas anteriores por medio de funciones matemáticas y las propagan a las siguientes asignando a su salida un valor coherente a la funcion que contiene y las entradas que recibió de otras neuronas. De esta forma aprende ajustando los valores que cada neurona toma en funcion de las conexiones, con el fin de minimizar el error en la etapa de clasificación. Un ejemplo de este tipo de estructuras son las redes neuronales usadas para clasificar imágenes pues estas pueden contener un gran numero de capas y en funcion que se acercan al final de la red las representaciones que generan son mas especificas de cada imagen procesada, así al explotar distintos niveles de características durante muchas iteraciones aprende a reconocer los patrones que definen una clase en particular permitiendo diferenciarla de otras [23].

2.2.1 Redes neuronales convolucionales

Las redes neuronales convolucionales como ejemplo claro de redes neuronales profundas son un conjunto de capas que extraen información de forma jerárquica, es decir, en funcion de los parámetros que definen cada capa y su ubicación en la red puede extraer información mas o menos relevante de esta forma una red neuronal convolucional o CNN por sus siglas en ingles puede distribuir las características a explotar de cada imagen en diversos grupos de capas de representación. Las capas de extracción de características y representación de este tipo de algoritmos esta compuesta por dos funciones básicas : convolución y agrupación o pooling. [23]

La convolución en redes neuronales profundas es una operación matemática que permite a través de unas matrices de parámetros o también llamados pesos que recorren un conjunto de valores numéricos los cuales representan las imágenes y generan un ponderadado para cada uno de estos valores en funcion de los valores asignados a sus vecinos. En procesamiento de imágenes, estas ponderaciones permiten obtener características predominantes en las imágenes como bordes verticales, horizontales, según la

configuración de estas matrices de parámetros o filtros. El resultado final es una matriz de nuevos valores con una dimensión que va a depender del tamaño de los filtros.

La operación pooling o agrupación va a utilizar parches o matrices de dimensión menor a la representación obtenida en la convolución, y se desplazara por estos mapas de características obtenidos buscando asignar un único valor que represente de forma coherente el conjunto de características como puede ser un máximo en un grupo de n valores o el promedio de este. Así reduce en dimensión la matriz de representación tras las convoluciones disminuyendo la complejidad computacional del cálculo sin perder información relevante. Las siguientes imágenes intentan explicar cómo se produce la convolución en aplicaciones de 2 dimensiones como la clasificación de imágenes. [23].

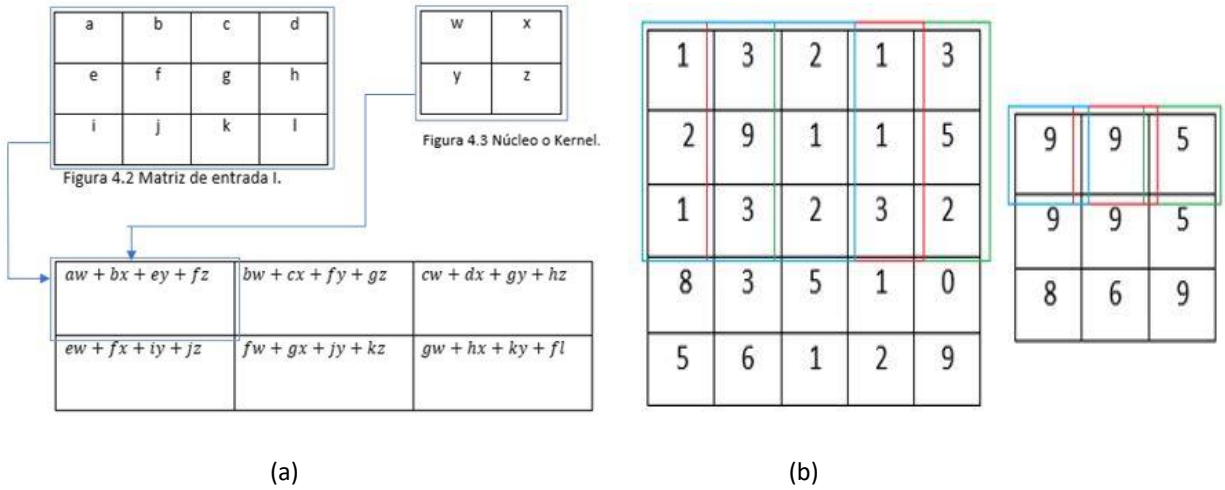

Figura 2.1 Ilustración grafica de operaciones convolución (a) y agrupación (b)

[Coursera clasificación de imágenes]

La figura 2.1 muestra de una forma simple como en operaciones de convolución el valor asignado a un espacio del mapa de características guarda relación con los pixeles vecinos, en tanto la funcion agrupación o pooling muestra como cada grupo de valores que recorre este núcleo o parche queda representado por el mayor valor, esta operación se llama MaxPooling y es muy utilizada en redes neuronales convolucionales.

2.3 Definiciones asociadas a redes neuronales convolucionales

En esta sección se mencionan una serie de conceptos que suelen ser muy representativos de este tipo de aprendizaje automático. Su descripción se hace necesaria debido a que juegan un papel muy importante

en la implementación de modelos de clasificación en librerías como Keras, Tensorflow, Pytorch entre otros.

2.3.1 Capas completamente conectadas (FC : Fully Connected)

En redes neuronales convolucionales las capas completamente conectadas están ubicadas usualmente en la etapa final de la red. Son un conjunto de neuronas agrupadas de forma secuencial y como su nombre lo indica guardan una conexión directa con cada una de las neuronas de la capa anterior y siguiente. La última capa completamente conectada también llamadas densas terminan con una dimensión igual al numero de categorias a clasificar. La conexión entre cada neurona y las neuronas vecinas esta controlada por un conjunto de parámetros llamados pesos sinápticos los cuales se actualizan al final de cada iteración con el fin de minimizar el error en la salida del modelo. En las capas completamente conectadas interactúan tres parámetros : un valor de entrada proveniente de la acumulación de neuronas anteriores, los pesos que regulan la salida de cada neurona y el sesgo visto como una variable estadística que busca estimar el error. [23]

2.3.2 Funcion de activación

Son funciones matemáticas usadas para controlar la salida de una neurona en funcion de los valores de entrada de las anteriores neuronas. Se espera que una neurona tenga una salida restringida por un limite inferior y superior en ocasiones en un rango de [-1, 1] o incluso [0, 1]. Tal como se explicó en la definición anterior de capas completamente conectadas, existen parámetros asociados a la salida de cada neurona y son en este caso el conjunto de parámetros que usados como entrada de la funcion de activación permiten obtener una salida normalizada. En redes neuronales se suelen usar funciones que, en caso de ser derivada, este proceso no sea demasiado complejo para la computadora. Además, si en la clasificación de imágenes se presentan transformaciones y procesos que no son lineales las funciones de activación deben estar acordes a estas condiciones. [18]

2.3.3 Propagación hacia atrás (back propagation)

La propagación hacia atrás permite el calculo de gradientes para encontrar los parámetros bajo los cuales se minimiza el error. La información en las redes neuronales convolucionales se transporta por ellas entre capas desde las capas inferiores hacia las superiores del algoritmo. Una vez obtenida una salida y calculado un error por medio de la etapa de validación, este error se convierte en la referencia para actualizar los parámetros de la red. Es matemáticamente similar a la regla de la cadena, derivando las funciones desde

la capa superior hacia las anteriores los gradientes calculados en esta etapa ayudaran a actualizar la red para la siguiente época o iteración.[23]

2.3.4 Métodos de optimización y tasa de aprendizaje

Estas dos definiciones son esenciales en conjunto con el proceso de propagación hacia atrás, los métodos de optimización en redes neuronales permiten obtener los nuevos pesos de la red minimizando el error en la salida a través de gradientes que ya se ha explicado como se obtienen. Por tanto, la tasa de aprendizaje medirá la proporción en la cual se actualizan los gradientes, usualmente es un valor pequeño en muchos ejemplos del orden de 1E-3 o inferior dependiendo de las técnicas implementadas para entrenar la red. Cuando el algoritmo converge hacia cierto valor de exactitud y el error en la salida es pequeño este valor suele ser reducido para evitar que grandes cambios afecten el calculo de gradientes y parámetros en las capas del modelo.[24]

2.4 Descriptores de características de locales

Los procesos de clasificación de imágenes usando descriptores de características locales están distribuidos en varias etapas : detección de puntos de interés, extracción de características y un proceso de representación de imágenes realizado por medio de algún método de clustering o agrupación y seguida una estrategia de codificación que permita obtener un vector adaptado para ingresar a algún método de clasificación como SVM o clasificador de bosques aleatorios. [25]

2.4.1 Detección de puntos de interés

Este proceso se trata de encontrar puntos o zonas en una imagen que puedan ser determinantes para definirla. La detección consiste en encontrar los puntos con detalles distintivos como cambios de contraste, bordes o esquinas, pero no es el punto como tal lo mas relevante sino las características extraídas de cada punto que definen matemáticamente esos contrastes, bordes, esquinas o cualquier otro rasgo distintivo de la imagen. Algunos de los descriptores usados en aplicaciones de vision por computador han sido Sift, Brisk, Hog, Brief o Harris Corner Detector por mencionar algunos.[20] [25].

2.4.2 Codificación por vectores de Fisher

La representación de imágenes a través de vectores de Fisher, la cual consiste en asignar a un conjunto de puntos de interés extraídos previamente un modelo de mezcla gausiana, buscando obtener las gaussianas que mejor representan cada conjunto de puntos. Se calculan las variables estadísticas que determinan la relación entre las características de cada punto de interés y la gaussiana que le ha sido

asignada. Basados en el principio de medir la participación de cada gaussiana sobre los puntos extraídos de la imagen, cada modelo inmerso en la representación de estos puntos tendrá asignadas unas variables como mediana, covarianza o desviación. El objetivo es medir la variación de los puntos asociados a cada gausiana en función de la posición en el grupo, es decir que tanto cambia la probabilidad de pertenecer a ese modelo ante la dirección de cambio respecto a los parámetros asociados a la gausiana (que puede representarse de forma general como elipses o círculos). Una vez se obtiene un conjunto de vectores como medidas de la variación respecto a la covarianza y la desviación, la concatenación de cada vector transpuesto en un solo arreglo de vectores será el vector de representación de la imagen.[25][20]

2.4.3 VLAD (Vector Locally Aggregated descriptor)

El objetivo de VLAD (Vector Locally Aggregated descriptor) es compactar la información extraída por medio de los puntos de interés desde K-Means hacia un vector que represente la imagen analizada. A diferencia del descriptor basado en vectores de Fisher, VLAD no utiliza modelos de mezcla gausiana para agrupar los puntos de interés y construir un diccionario de palabras, en su lugar utiliza el agrupamiento por K-Means y el centroide de cada uno de sus clústeres o agrupaciones como parámetro para la asignación y construcción de histogramas. En el caso de VLAD, la medida de variación será la distancia entre punto y centroide manteniendo el signo del cambio, con el fin de conocer la dirección respecto al centroide.

Al sumar para cada grupo de puntos las distancias calculadas, se obtiene un vector de una dimensión para cada agrupación. La concatenación de todos los vectores de distancia resultantes genera el vector de representación local VLAD. .[25][33]

2.4.4 Bag of visual words (BOVW)

La bolsa de palabras visuales es un método de representación de imágenes similar a VLAD en el sentido que usa el método de agrupación K-Means para dividir las características extraídas en los puntos de interés. Una vez asignado un centroide a cada descriptor extraído de los puntos de interés, se genera un vocabulario donde cada elemento de este corresponde a un centroide calculado por medio de K-Means. Cada miembro del vocabulario o palabra tiene asignado un histograma que representa la frecuencia con la cual se repite una característica en el clúster. El vector codificado contiene información sobre la presencia las distintas características en cada palabra a través de los histogramas. Al relacionar las palabras con las imágenes, cuando una imagen contenga descriptores con frecuencias similares para cada característica existe la probabilidad de que estas imágenes pertenezcan a una misma categoría.[20].

2.4.5 Maquina de soporte vectorial (SVM)

Una máquina de soporte vectorial es un método de clasificación formado por un hiperplano separador capaz de discriminar clases en un conjunto de puntos. Dado un conjunto de datos etiquetados el algoritmo genera un hiperplano que permite diferenciar entre los datos de una etiqueta y otra. En un problema de 2 dimensiones, la línea que discrimine óptimamente dos conjuntos de puntos será aquella que pase a una distancia suficiente de los dos grupos para superar la sensibilidad al ruido y mejorar la discriminación. En el caso de múltiples dimensiones, el problema consiste en generar un hiperplano a partir de la mayor distancia posible a los datos de entrenamiento. Dos veces el valor de esta distancia recibe el nombre de margen. El máximo margen posible es el valor optimo del algoritmo. Este tipo de problemas se encuentra en la categoría de clasificación lagrangiana, en la cual el objetivo está en maximizar el margen y las restricciones están sujetas a la discriminación de clases por medio del hiperplano. [26**]

2.5 Software y librerías especializadas

Para llevar a cabo tareas de clasificación de imágenes, reconocimiento de objetos, segmentación o cualquier otra tarea dentro del campo de vision por computador es necesario algunas de las herramientas que pueden facilitar el desarrollo de los programas debido a su gran cantidad de funciones. En esta sección se mencionaran algunas de las librerías y entornos de programación necesarios para la implementación de modelos clasificadores de imágenes o descriptores locales de características en este caso.

2.5.1 Google Colab

Es un entorno de programación basado en Python de acceso gratuito a través de la cuenta de Google, el cual permite acelerar cálculos complejos gracias al acceso libre de GPU (Unidad de procesamiento gráfico) al igual que la posibilidad de ejecutar rutinas computacionalmente exigentes en Tensorflow pudiendo acelerar los procesos por medio de TPU (Unidad de procesamiento tensorial) conectándose a un entorno virtual de Google.[28]

2.5.2 OpenCV

Esta biblioteca de código abierto usado de forma popular a través de Python contiene algoritmos optimizados en el campo del aprendizaje automático y vision por computador. Entre las aplicaciones de sus algoritmos esta la detección y objetos, clasificación de imágenes, procesamiento de imágenes y video lo que permite el desarrollo de aplicaciones que funcionan en tiempo real debido a que sus algoritmos facilitan interactuar con cámaras y detectar movimiento en ellas, entre muchas otras opciones. Además

de contar para este caso especifico con algoritmos capaces de detectar características locales en imágenes de forma similar a los algoritmos desarrollados en VLFeat.[20]

2.5.3 Pytorch

Pytorch es una biblioteca de inteligencia artificial optimizada para el funcionamiento tanto en CPU (unidad de procesamiento central) como en unidades de procesamiento gráfico. Sus funciones están enfocadas en el aprendizaje profundo por lo que es posible diseñar redes neuronales artificiales para diferentes propósitos. Debido a que se enfoca principalmente en cálculos a partir de tensores permite interactuar con librerías matemáticas como numpy y sustituir algunas de sus funciones.

2.5.4 Tensorflow – Keras

Tensorflow es una librería de inteligencia artificial basada en operaciones tensoriales, como se había explicado para Pytorch con la posibilidad de usarse en conjunto con unidades de procesamiento gráfico. Muy usada en conjunto con Keras para el diseño de redes neuronales secuenciales. Los modelos de redes neuronales, optimizadores, funciones de activación y construcción de capas o layer cuando se ejecuta sobre TensorFlow permite construir aplicaciones de forma intuitiva. [32]

2.5.5 VLfeat

Esta librería de vision por computador tiene un enfoque similar a OpenCV. Su potencial se encuentra en la explotación de características de las imágenes, y procesamiento de descriptores. Debido a que se centra en extracción de características y representación de imágenes contiene aplicaciones desde detección de puntos de interés, hasta algoritmos capacitado para realizar agrupación de datos como K-Means o GMM (Gauss Mixture Models), además de aplicaciones para codificar los datos agrupados por las anteriores opciones como VLAD o vectores de Fisher. Diseñado para la implementación de sus aplicaciones Matlab.[35]

Capítulo 3. Metodología

3.1 Selección de imágenes de entrenamiento

En la selección de las bases de imágenes para las etapas de entrenamiento, validación y prueba, las imágenes deben estar organizadas de forma tal que al ingresar al algoritmo de aprendizaje se pueda reconocer la clase, dirección en el sistema de almacenamiento o path, y la tarea para la cual están destinadas. Para este caso particular se usó la siguiente estructura:

	Numero de imágenes por clase		
Base de imágenes	Entrenamiento	Validación	Prueba
JUNIO20V1	100	25	25
FEBR20V3	100	25	25
JUNIO20V3	100	25	25
FEBR20V2	100	25	25
FEBR19V3	100	25	25
FEBR19V2	100	25	25
FEBR19V3IMAG18	100	25	25
15-SceNe	100	10	15
Sports	70	35	25

Tabla 1 Distribución de datos de entrenamiento para clasificación con redes neuronales convolucionales

De forma alternativa se realizó una distribución en la cual los datos del proceso de validación fueron usados en su totalidad como datos de prueba, es decir aplicados a las funciones de predicion una vez el modelo sea entrenado satisfactoriamente.

Para la tarea de clasificación de escenas se considera una escena como una imagen en la cual intervienen un conjunto de objetos ubicados en distintas partes de una imagen, los cuales se caracterizan por estar presentes espacialmente en una gran cantidad de imágenes de la misma categoría. Es importante tener un gran número de imágenes que varíen en la ubicación, cantidad y tipo de objetos que la componen de forma que sea posible identificar patrones similares en imágenes que no hayan sido vistas antes por el algoritmo. Cabe aclarar que la preparación y distribución de los datos varía dependiendo tanto del tipo de tarea como del tipo de aprendizaje automático que se piensa implementar [23].

3.2 Preparación de bases de imágenes en tareas de clasificación de escenas.

Las imágenes de entrenamiento serán un conjunto de escenas agrupadas en clases las cuales serán la base para extraer características por medio de las capas convolucionales de la red neuronal artificial, estas pueden estar sometidas a transformaciones aleatorias que buscan entrenar la red con una variedad de escenarios distintos evitando así que el algoritmo aprenda a reconocer únicamente rasgos muy específicos de las imágenes de entrenamiento y muestre resultados negativos al aplicarse a escenarios distintos a este. El segundo grupo llamado validación se encarga de verificar el error en la etapa de entrenamiento, este contiene una cantidad de imágenes generalmente menor a las de entrenamiento e idealmente se espera que conserven similitud con las anteriores dado que el resultado de aplicar filtros convolucionales y de agrupación a las imágenes de validación permitirá ajustar los pesos sinápticos de la red minimizando el error en el proceso de entrenamiento del algoritmo. En último lugar se encuentran las imágenes de prueba están son una muestra de imágenes que el algoritmo no ha visto ni procesado en las etapas de entrenamiento y validación, se usan para generar predicciones a partir del modelo de aprendizaje automático generado y evaluar su desempeño [9][15].

Las imágenes usadas en la implementación de redes neuronales para las tareas de clasificación de escenas están compuestas por imágenes variadas en su composición, es decir, La cantidad de objetos, partes de objetos, o colecciones de objetos varían a través de las diferentes bases de imágenes, debido a que algunas clases en una base de imágenes pueden ser el resultado de la modificación de imágenes pertenecientes a otra base. A continuación, se presenta una tabla explicativa de las diferentes características presentes en las bases de imágenes:

Nombre	Tipo de entrenamiento	Tipo de validación	Tipo de prueba
Escenas	Imágenes de escenas	Imágenes de escenas	Imágenes de escenas
JUNIO20V1	Imágenes de objetos y partes distorsionadas	Imágenes de escena	Imágenes de escenas
FEBR20V3	Imágenes de escenas	Imágenes de objetos	Imágenes de objetos
JUNIO20V3	Imágenes de escenas	Imágenes de objetos	Imágenes de objetos
FEBR20V2	Imágenes de objetos	Imágenes de escenas	Imágenes de escenas
FEBR19V3	Imágenes de escenas y objetos	Imágenes de escenas y objetos	Imágenes de escenas y objetos

Nombre	Tipo de entrenamiento	Tipo de validación	Tipo de prueba
FEBR19V2	Imágenes de objetos	Imágenes de objetos	Imágenes de objetos
FEBR19V3IMAG18	Imágenes de objetos	Imágenes de objetos	Imágenes de objetos
15-Scene	Imágenes de escenas a color y a blanco y negro.	Imágenes de escenas a blanco y negro	Imágenes de escenas a blanco y negro
Sports	Imágenes de escenas	Imágenes de escenas	Imágenes de escenas

Tabla 2Características de las bases de imágenes a utilizar.

Si bien el ideal para entrenar una red neuronal sin pesos ni estadísticas previas aprendidas es una gran cantidad de datos en general cientos de muestras por categoría, para este caso particular en el cual usaremos las características básicas aprendidas en reconocimiento de escenas de arquitecturas como Resnet-50 o VGG-16 nuestros conjuntos de muestras serán particularmente reducidos pero suficientes para obtener buenos resultados. La siguiente grafica muestra la distribución en número de muestras para cada categoría [3][17].

A continuación, se presenta gráficamente la forma en que están distribuidos los datos a utilizar, con el fin de utilizar la relación entre el número de muestras en las bases propias y las bases de imágenes que son subconjuntos de bases usadas habitualmente en el campo del reconocimiento de escenas tales como MIT Indoor, Places365, Caltech101, entre otras

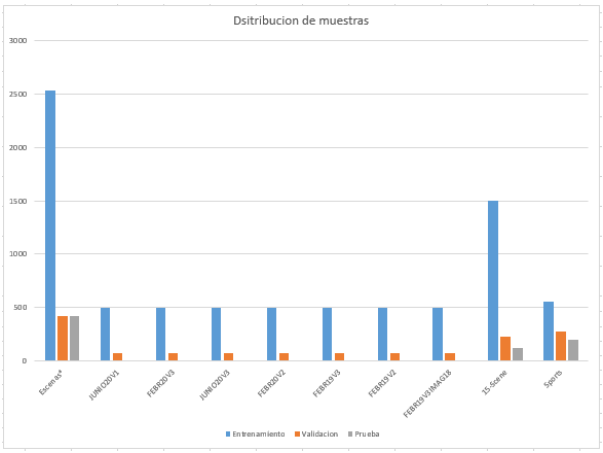

Figura 3.1 Distribución del número de muestras por etapas en el algoritmo para las 10 bases de imágenes.

3.3 Tipos de bases de imágenes

De acuerdo a la aplicación que se desee llevar a cabo con una red neuronal artificial se elige el tipo de datos de entrenamiento a utilizar, por poner un ejemplo cuando se desea detectar objetos en una

fotografía o video las muestras de entrenamiento del algoritmo deberán tener identificadas por medio de etiquetas el área y ubicación del objeto en la imagen, usando herramientas como LabelImg, LabelMe o IBM cloud annotations para realizar etiquetado y creación de archivos tipo json requeridos para algoritmos de detección como Single Shot Multibox SSD. Pero también es necesario comprender la preparación de datos de entrenamiento para algoritmos de clasificación de imágenes en los cuales se requiere la agrupación de fotos por etiqueta en las cuales toda la imagen representa una muestra real de la clase a clasificar. [25][3]

Con el fin de preparar las imágenes anteriormente descritas para cualquiera de los fines mencionados en esta definición, se presentarán muestras etiquetadas tanto de forma rectangular para detección de objetos como el útil etiquetado poligonal para entrenar algoritmos de segmentación de objetos.

3.4 Etiquetado por cajas delimitadoras / etiquetado rectangular

LabelMe permite crear múltiples etiquetas en una misma imagen asignando colores específicos a cada objeto clasificable además cuenta con scripts que permiten la fácil conversión de la información en etiquetas a distintos formatos lo que permite llevar bases de entrenamiento disponibles para algoritmos SSD a algoritmos de detección como YOLO o similares. [3] La base de imágenes FEBR19V3IMAG18 cuenta con cinco clases distintas, algunas de ellas con objetos comunes entre si lo que permitirá evaluar de mejor forma el rendimiento del algoritmo de aprendizaje automático. En resumen, FEBR19V3IMAG18 cuenta con las siguientes clases.

CLASE 1: La clase 1 también denominada 'Cocinas' es una colección de objetos entre los que se encuentran sillas, mesas, globos, banderas, TV, equipos de sonido que tienen como particularidad el no aparecer entre las otras categorías.
CLASE 2 y CLASE 3: son imágenes de salas de estar e imágenes propias de un hogar típico, con objetos sobrepuestos y límites de difícil demarcación.

CLASE 4 y CLASE 5: Son imágenes propias de la universidad tecnológica de Pereira, cuenta con escenas típicas de un establecimiento educativo: oficinas, carteles, puertas, grandes ventanas, pasillos y casilleros. La diferencia entre ambos esta entre los objetos a etiquetar en una y otra clase dado que comparten espacios comunes en ambos casos.

A continuación, se presenta una muestra de las imágenes etiquetadas por medio de la aplicación LabelMe cuando se requiere crear una base de imágenes para algoritmos de detección de objetos.

(a) (b) (c)

Figura 3.2 Etiquetado de imágenes por LabelMe

Las imágenes anteriores son ejemplos de la clase 1 de la base de imágenes FEBR19V3IMAG18 en la cual se ilustra la forma en la cual se realiza el proceso de etiquetado encerrando en áreas rectangulares identificando con cada color la etiqueta definida para un objeto. En este tipo de programas la tarea es completamente manual teniendo que resaltar el área por objeto a través de cientos de imágenes [3].

3.5 Etiquetado poligonal para aplicaciones de segmentación.

Este tipo de etiquetado es útil para algoritmos de segmentación de imágenes en los cuales se busca detectar áreas y siluetas de objetos específicos. En este tipo de aplicaciones el algoritmo es entrenado a partir de los contornos permitiendo aprender no solo los patrones de una porción del objeto sino además su forma física dentro de la imagen. Algunos ejemplos de este tipo de aplicaciones se ven a continuación:

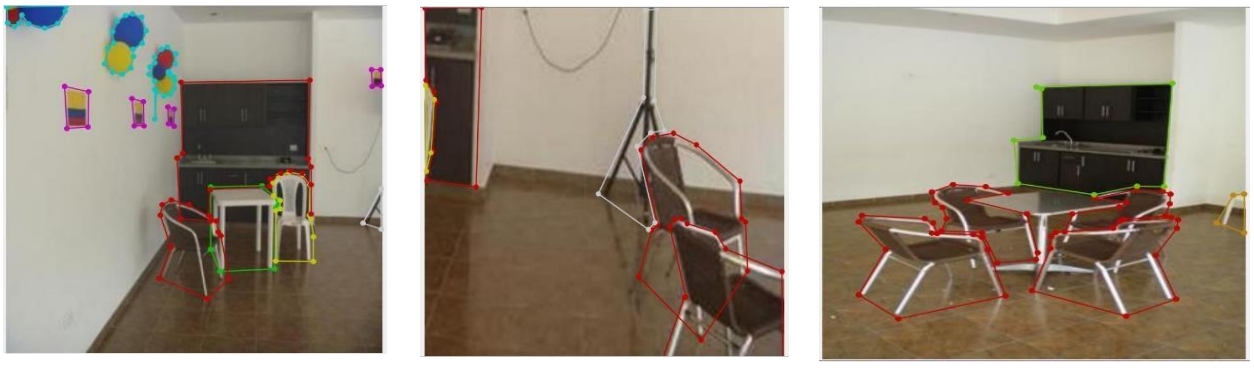

Figura 3.3 Etiquetado poligonal por LabelMe

3.6 Etiquetado semiautomático – IBM Cloud Annotations

Otra de las herramientas exploradas en la preparación de datos de entrenamiento es la aplicación de etiquetado semi automático desarrollada por IBM llamada Cloud Annotations su funcionamiento se basa en algoritmos de detección de objetos, en este caso los objetos detectados son convertidos en etiquetas de las imágenes ingresadas. Conociendo la filosofía del aprendizaje supervisado si requerimos etiquetar una gran cantidad de imágenes se debe dar una muestra de etiquetas correctas las cuales se realizan manualmente siendo 20 el número mínimo de objetos por etiqueta ingresados manualmente para la ejecución del algoritmo sobre las imágenes restantes. Para este caso se ingresaron imágenes de muestra como las vistas en etiquetado rectangular arrojando de forma automática los siguientes resultados [25]:

Figura 3.4 Resultados de etiquetado semiautomático de IBM Cloud Annotations

Las etiquetas obtenidas pertenecen a algunos de los objetos manualmente asignados, este tipo de etiquetado está sujeto a las variaciones en las muestras ingresadas manualmente lo que mejorara o disminuirá su rendimiento final en la etapa de detección de la aplicación Cloud Annotations [25].

3.7 Selección de muestras para clasificación de imágenes

En este apartado se visualiza un pequeño número de imágenes por cada categoría que compone la base de imágenes FEBR19V3IMAG18, donde se pueden observar similitudes entre algunos grupos de imágenes. El total de datos de entrenamiento es de 500 imágenes dividas en 100 imágenes por clase, para la tarea de validación se toman grupos de 25 imágenes por clase para evaluar el desempeño .

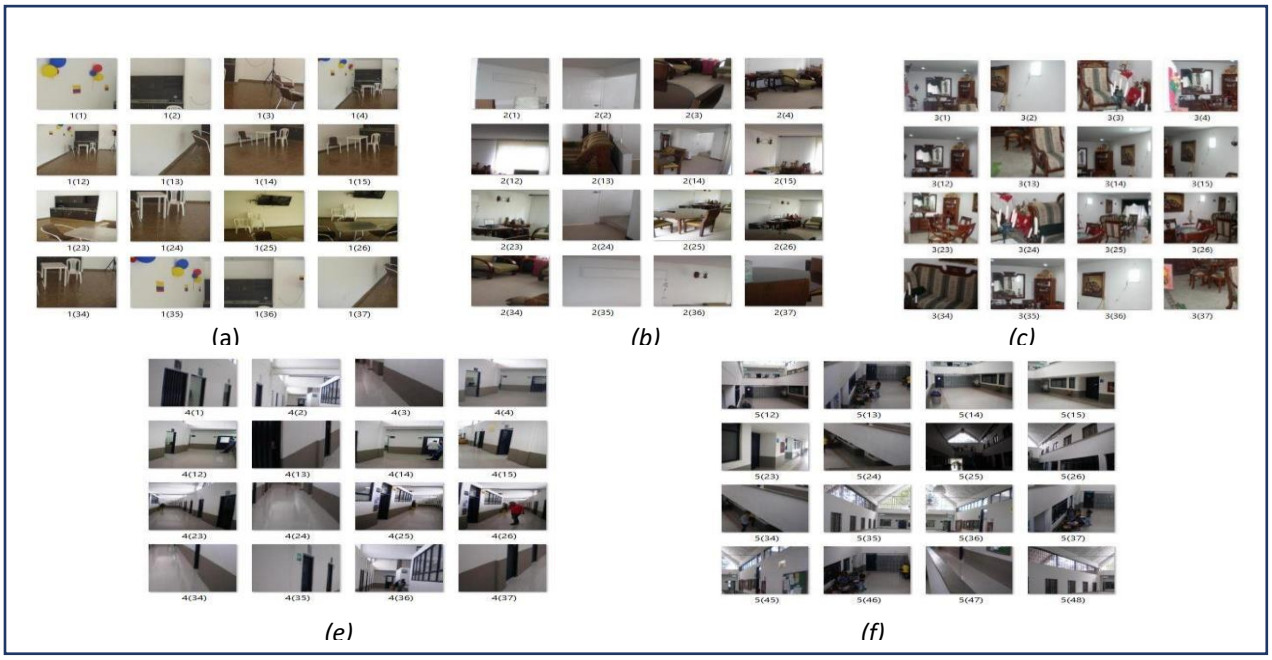

Figura 3.5 Presentación de muestras aleatorias a utilizar en el algoritmo de aprendizaje

Las imágenes mostradas representan una muestra de las imágenes de entrenamiento agrupadas según las clases a clasificar donde se pueden observar similitudes entre algunos grupos de imágenes. El total de datos de entrenamiento es de 500 imágenes dividas en 100 imágenes por clase, para la tarea de validación se toman grupos de 15 imágenes por clase y 10 imágenes de cada grupo se tomarán para evaluar el desempeño del algoritmo. Estas 10 últimas imágenes no son vistas por la red neuronal hasta la etapa de evaluación.

3.8 Datos aumentados

La técnica de datos aumentados nace de la necesidad de recapturar datos de una misma escena al tiempo poder presentar aleatoriamente distintas posiciones y enfoques de las imágenes que eviten que la red aprenda de una forma muy específica y no pueda funcionar correctamente cuando se deben procesar imágenes desconocidas. Este método genera de una misma imagen rotaciones, recortes y enfoques algo que funciona bien cuando no se cuenta con una gran cantidad de datos de entrenamiento [24][9].En las imágenes siguientes se puede observar los distintos efectos y enfoques que pueden generarse aleatoriamente en las imágenes de entrenamiento, además el método sugiere dejar sin alteraciones las imágenes de validación de esta forma el algoritmo aprende a reconocer objetos independientes de la orientación o el tamaño de la imagen [24]

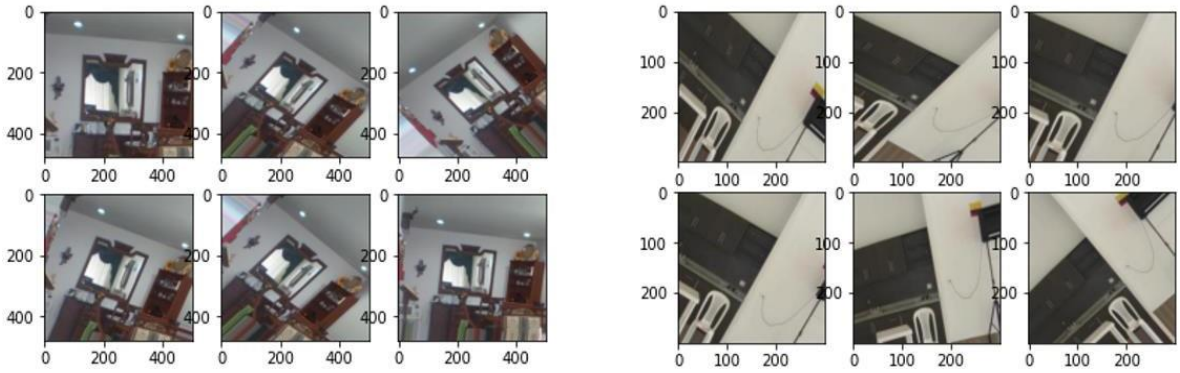

Figura 3.6 Visualización de datos aumentados en clases de FEBR19V3IMAG18

3.9 Entrenamiento y evaluación de clasificadores de imágenes usando redes neuronales

CASO 1.1 : VGG16 (Visual Geometry Group de 16 capas).

Con el fin de evaluar el rendimiento en modelos de clasificación de imágenes previamente entrenados, se importa a través del módulo *keras.applications( )* la arquitectura VGG16. Esta arquitectura alojada junto un grupo de redes neuronales pre entrenadas tuvo un gran éxito en la competencia Imagenet ILSVRC y sus parámetros están de hecho ajustados por defecto sobre esta base de imágenes. Imagenet contiene aproximadamente 1,2 millones de imágenes agrupadas en cerca de 1000 características. Inicialmente la red importada contiene 13 capas convolucionales y 3 capas completamente conectadas, sobre la cual inicialmente la única modificación será variar el número de categorías a clasificar. Debido a que las bases de imágenes propias contienen entre 5 y 15 categorías según el caso, que además de ser números mucho menores a los del modelo original no coinciden con los subconjuntos de imágenes con los cuales fue entrenado inicialmente el modelo VGG16-Imagenet, se hace necesario modificar por completo la capa de clasificación.

Una vez aclaradas estas condiciones, se procede a entrenar la red neuronal convolucional (CNN por sus siglas en inglés) exponiendo inicialmente las imágenes de la base *FEBR19V3IMAG18* la cual está compuesta por cinco categorías que incluyen fotografías de salas, cocinas y pasillos, todos elementos agrupados en la categoría *escenas de interiores,* la cual es el objeto de estudio de las metodologías a presentar. Durante 100 iteraciones, un rango de resolución de imagen no superior a 224x224 (ancho por largo en pixeles), una tasa de aprendizaje sugerida de 1E-3 (0,001) y 100 imágenes de entrenamiento por clase frente a 25 en la etapa de validación y prueba se realiza el entrenamiento del algoritmo usando como elemento optimizador de gradientes el método RMSProp. Durante esta etapa los parámetros de la red estarán congelados, es decir, los filtros convolucionales actuaran como extractores de características usando parámetros aprendidos durante su prueba en la competencia Imagenet ILSVRC y será en las capas completamente conectadas con salidas de dimensión 5, 8 o 15 categorias donde se ajuste el modelo a las imágenes contenidas en *FEBR19V3IMAG18.* Esta práctica que va desde el entrenamiento de la red neuronal hasta la obtención de la probabilidad de acierto por clase será definida como modelo base y será el punto de partida para las demás practicas propuestas.

CASO 1.2. Aplicación de pesos pre entrenados en Places365 al modelo del caso base.

Una vez obtenida tanto la exactitud en general del clasificador para cada base de imágenes como la exactitud del modelo sobre cada clase puesta a prueba, se procede a realizar la carga de un nuevo conjunto de parámetros. Para este nuevo caso se importará desde la plataforma GitHub especialmente desde el perfil de usuario perteneciente a Grigorios Kalliatakis el conjunto de parámetros pre entrenados en Place365. Este conjunto de datos es una herramienta de acceso libre asociada al Instituto Tecnológico de Massachusetts (MIT) específicamente al grupo de vision por computador del mismo instituto. El conjunto de datos está dividido en dos grupos, un subgrupo llamado Places365-Standard el cual contiene 1.8 millones de imágenes de entrenamiento y un grupo superior llamado Places365-Challenge 2016 con 8 millones de imágenes de entrenamiento que incluyen las imágenes asociadas a Places365-Standard. Como su nombre Places365 lo indica contiene 365 categorias distintas de escenas. Como explicamos anteriormente en el capítulo 1, una escena es una imagen con un conjunto de objetos representativos de un lugar en particular como pueden ser lugares interiores o exteriores. La elección de este conjunto de imágenes como base de los parámetros previamente entrenados se debe a la similitud de algunas de las 365 categorias con las imágenes propias en la lista de bases enunciadas en el caso I.

Teniendo en cuenta lo anterior, se aplica exactamente el procedimiento mencionado en el caso I. Es decir, al reemplazar el conjunto de pesos por defecto Imagenet por Places365 por medio de la

herramienta alojada en Tensorflow *model.load_weights('path_parametros')* donde path_parametros corresponde al archivo obtenido desde el perfil de GitHub mencionado. Esta será la única variación con respecto al caso I dado que el objetivo sigue siendo el mismo, obtener la exactitud general del modelo y el porcentaje de aciertos para cada clase en el conjunto de validación y prueba.

CASO1.3 Descongelación de capas convolucionales para VGG16

Ya obtenidos los resultados de clasificación por medio de la arquitectura de red neuronal convolucional VGG16 usando como base tanto los pesos obtenidos desde Imagenet como los obtenidos desde Places365, se procede a descongelar algunas capas del conjunto de 13 capas convolucionales para ser entrenadas de nuevo junto a las capas complementamente conectadas que actúan como parte del nuevo clasificador. Descongelar una capa implica que los parámetros y estadísticas asociadas a ella desde el momento en que se obtuvo el modelo serán inicializados aleatoriamente, es decir, tendrán un valor inicial que no hace parte del conjunto de pesos importados. Al visualizar la estructura de la red por medio de Google Colab se observa que estas capas estarán sujetas a un nuevo entrenamiento y por tanto se sugiere modificar la tasa de aprendizaje con el fin de que la actualización de los nuevos pesos en las capas finales a partir de las características extraídas no perjudique en gran forma el rendimiento del modelo base. Para el caso de esta última variación la tasa de aprendizaje fue disminuida de 1E-3 a 1E-4 es decir pasó de 0.001 a un valor de 0.0001 por motivo de la explicación anterior.

Como Argumento para elegir que capas serian reinicializadas en busca de mejoras en el rendimiento de la red, se tomó el hecho de que el ultimo bloque convolucional de una CNN extrae características muy específicas de los datos expuestos a entrenamiento, es decir, en el paso del cuarto al quinto bloque convolucional el mapa de características que representa la imagen ya ha incluye detalles genéricos de esta y se ha formado una representación significativa. Si se modifican las capas asociadas al último bloque convolucional es posible aprender a reconocer patrones muy representativos de las imágenes propias y en conjunto con las características aprendidas en las capas anteriores podría presentar en algún caso mejoras de rendimiento. Una vez comprendido esto, por medio de las instrucciones *layer.trainable, set_trainable y model.trainable* los cuales reciben valores booleanos True(verdadero) o False(falso) es posible asignar instrucciones de entrenamiento, reentrenamiento o reinicializacion de pesos a cada componente de la CNN. Para este caso elegimos explorar el reentrenamiento de las capas nombradas en el modelo VGG16 de Keras como *conv5.3, conv5.2* o el conjunto de *conv5.3* más *conv5.2.* En forma resumida los únicos parámetros que se ajustarían de nuevo en función de las características aprendidas serían los del bloque final de clasificación

conformado por las capas completamente conectadas junto a la capa de salida con activación tipo softmax y el conjunto conformado por la ultima y penúltima capa convolucional.

Finalmente se repite la parte final de cada etapa de entrenamiento en el algoritmo, pasadas el número de épocas o iteraciones predefinidas es posible visualizar los resultados y medir el rendimiento del modelo en función del número de errores y aciertos siendo la exactitud (accuracy en la implementación del algoritmo) el número total de aciertos sobre el número total de muestras evaluadas.

CASO 1.4 Clasificación de imágenes usando Resnet50

Para el caso 1.4 Se elegio la arquitectura de CNN Resnet50 la cual presenta una mayor profundidad en cuanto al número de capas. Al igual que en casos anteriores, entrenar una red desde cero que requiere calcular y ajustar de forma progresiva millones de parámetros requiere una gran cantidad de imágenes por tanto se hace uso de un modelo de red neuronal como este, pero previamente capacitado para clasificar con un buen nivel de exactitud cerca de 1000 clases distintas vista desde Imagenet o más de 300 escenas vista desde Places365. Para este caso y dado que estaba evaluada la red VGG16 tanto en Imagenet como en Places365 se opta para entrenar el algoritmo directamente a partir de la base de 365 escenas de MIT. La razón, como se vio en el caso 1.2 (VGG16 usando Places365) una red neuronal convolucional que se ha capacitado para detectar características en escenas de interiores y que además posee un mayor número de capas para el procesamiento de imágenes y extracción de información podría funcionar mejor con el tipo de imágenes propias las cuales comparten cierta similitud.

Para este caso, se hace uso de un paquete de Python llamado Pytorch el cual permite realizar cálculos numéricos de gran complejidad acelerando estos procesos a través de unidades de procesamiento grafico si están disponibles y también permite la construcción de algoritmos basados en redes neuronales. Debido al tiempo y recursos computacionales que exigía el entrenamiento de redes neuronales para clasificación de imágenes se hace uso de un entorno de programación basado en Python disponible por medio de Google llamado Google Colab o Colaboratory para agilizar los procesos tanto para los primeros tres casos centrados en la utilización de VGG16 como para la utilización de Resnet50. La metodología en Resnet50 es en esencia la misma de los casos anteriores salvo el entorno en el cual se desarrolla, conservando los mismos tipos de optimizadores, tasas de aprendizaje, bases de imágenes a procesar y buscando el mismo objetivo obtener una exactitud general por cada base de imágenes al igual que la medida de exactitud por cada clase que las contiene. Siguiendo las mismas reglas del caso 1.3 descongelación de capas convolucionales se identifican las

partes de la red haciendo una lectura de su arquitectura e identificando la etapa final de esta, que en esta ocasión comparte un bloque convolucional final similar al caso de VGG16 solo que se denomina conv4.1, conv4.2 y conv4.3. En este caso se ejecuta un caso base sin ninguna modificación y se reinicializan conv4.2 y conv4.3 en todas sus combinaciones posibles. Finalmente se consignan los resultados con el fin de poder comparar los rendimientos para cada prueba.

3.10 Clasificación de imágenes usando descriptores de características locales

Con el fin de comparar los resultados de un descriptor global de características como puede interpretarse la etapa final de una CNN con los resultados posibles a partir de la extracción de características locales se implementan dos métodos muy usados en esta segunda parte : VLAD (Vector Locally Aggregated Descriptor) y BOVW (Bag of Visual Word). Tanto VLAD como la bolsa de palabras visuales traducido al español comparten una estructura similar de procesar la información. Primero se requiere la lectura de las bases de imágenes a procesar, acto seguido la elección de un método de extracción de características como SIFT, BRIEF, BRISK, HOG, entre otros. Una vez se selecciona el método de extracción, se procede al cálculo de estadísticas para cada zona o punto de la imagen ubicado y procesadas todas las imágenes, cada conjunto de datos asignado a una clase en particular se hace uso de un método de agrupación o clustering. Una vez agrupados los datos en clústeres se inicia un proceso de codificación de la información agrupada en la cual cada elemento de la matriz o vector resultante tendrá asignados un conjunto de valores que lo relaciona con el grupo o clúster asignado. Obtenido este vector de información y teniendo en cuenta que la información extraída de cada punto de interés está ligada a una clase de la base de imágenes, se utiliza una máquina de soporte vectorial para el cálculo de una frontera que divida cada clase dentro del conjunto de puntos y posteriormente se puedan realizar predicciones en función de este modelo de clasificación.

CASO 2.1 Clasificación de imágenes usando BOVW

Para el clasificador de imágenes a partir de bolsa de palabras visuales o BOVW se usa Matlab como entorno de programación haciendo llamados a las funciones de las librerías especializadas en vision por computador. Sigue tal como se mencionó inicialmente la misma secuencia usada para la implementación de VLAD, es decir una etapa de lectura de imágenes, extracción de características, agrupación , representación y posteriormente clasificación. Para el caso de BOVW la representación se realiza por medio de histogramas que definen la relación de cada característica con el centro de cada agrupación generada. A continuación, se describe de forma corta cada una de las etapas.

2.1.1. Lectura de imágenes

Para leer las imágenes a procesar se requiere identificar el directorio en el cual se encuentra la carpeta de 'TRAIN' y 'TEST' para cada base de imágenes. En este caso dado que se requiere generar un grupo de imágenes por cada clase se particiona el grupo total de imágenes en 5, 8 o 15 según las imágenes a clasificar. De esta forma se asigna de forma manual tanto las categorias en el proceso de entrenamiento como las categorias en el proceso de validación. A través de las funciones *subplot( ) y imshow( )* es posible visualizar muestras de las imágenes cargadas en el programa.

Para el clasificador basado en BOVW (bag of visual words) se usaron las siguientes bases de imágenes:

- JUNIO20V1
- FEBR20V3
- FEBR20V2
- JUNIO20V3
- 15-SceNe
- Sports

2.1.2. Detección de puntos de interés

Una vez obtenidas las rutas de imágenes y divididas en grupos de muestras iguales, el siguiente paso es la aplicación de un método de detección de puntos de interés como SIFT, el cual genera 128 características por cada punto detectado. El procedimiento es iterativo generando por cada imagen un conjunto de vectores de características que contienen en este caso valores en función de las variaciones en cada imagen como, por ejemplo, cambios de contraste, bordes, entre otros. Se genera un conjunto de vectores apilados hasta completar el total de las imágenes a explorar.

2.1.2. Agrupación de características

Obtenido el conjunto de características por cada punto de interés, por cada imagen, por cada clase se realiza el llamado de una librería de clustering, en este caso k-Means y se define inicialmente un numero de grupos a asignar sobre el conjunto de características obtenidas en el paso anterior. En este punto se obtendrá el centroide de cada grupo, de forma tal que se me minimiza la distancia entre el centroide y cada uno de los datos del conjunto de características. El número de agrupaciones se encuentra de forma experimental según los resultados obtenidos con distintos valores, mientras que las asignaciones entre las características agrupadas y cada centroide representante de un grupo se

realiza internamente en las funciones asignadas a K-Means sea en Python o en Matlab como es este caso.

2.1.3. Representación y clasificación

Una vez generados los grupos y asignados a cada característica extraída de la imagen se obtienen lo que se denominan palabras visuales e histogramas, cada palabra corresponde a un grupo de características que suelen tener alguna medida común, en el caso de K-Means es la distancia entre el centroide y cada punto del conjunto. El histograma es como tal la medida de las distintas características extraídas de la imagen y posteriormente agrupadas en relación con la palabra asignada. De cierta forma si cada clúster o agrupación luego de aplicar K-Means comparten características comunes en sus miembros el histograma podría verse como la representación de la imagen en función al centroide que fue asignado. Cuando se han realizado una serie de operaciones como normalizar los valores de los histogramas y codificados por medio de la función *encode( )* del código implementado se puede aplicar un algoritmo de clasificación como Maquina de soporte vectorial (SVM por sus siglas en ingles) para obtener finalmente una matriz de confusión que permite calcular tanto la exactitud en general del algoritmo sobre cada base de imágenes como la exactitud por cada clase y el número de imágenes clasificadas de forma incorrecta en cada una de las categorias restantes.

CASO 2.2 Clasificación de imágenes usando VLAD

Para el caso de VLAD Se sigue un procedimiento similar en el sentido de buscar la generación de un vocabulario de palabras y conserva la misma herramienta de Matlab para agrupar las características extraídas de la imagen, en este caso K-Means de las librerías de vision por computador. De igual forma se hace necesaria la definición de un numero de agrupaciones de forma predefinida que para este caso será 64, y un extractor de puntos de interés el cual seguirá siendo SIFT con 128 características por cada punto calculado. Para leer las imágenes y pre procesarlas antes de extraer los puntos de interés es necesario dividir el conjunto total de imágenes en intervalos de forma tal que cada intervalo corresponda al número de imágenes de una misma clase. El procedimiento aplicado para las imágenes de entrenamiento será igual al aplicado para las imágenes de validación dividiendo cada intervalo en 100 imágenes de entrenamiento por 25 de validación construyendo una estructura de imágenes igual a la usada en la clasificación por redes neuronales convolucionales.

Las imágenes bases de imágenes a utilizar en este caso serán las primeras cuatro categorias usadas en el clasificador de imágenes con BOVW, es decir :

- JUNIO20V1

- FEBR20V3
- FEBR20V2
- JUNIO20V3

Una vez leídas las imágenes desde Matlab, se procede a transformar el color a escala de grises como paso previo a la aplicación de la función *vl_sift( )* la cual genera información sobre los puntos de interés obtenidos y el vector de características de cada punto. Cuando esta función proveniente de la librería de código abierto VLFEAT se aplica sobre todas las imágenes almacenadas en la lista de entrenamiento desde la imagen 1 hasta 500, se obtiene un descriptor de características locales calculado para cada imagen. Este procedimiento se repite con las imágenes de validación de forma que se cuenta al final con un descriptor de características de entrenamiento y un descriptor de igual forma para los datos de validación. A diferencia del proceso de agrupación y asignación de centroides en este caso se hace uso principalmente de la función K-Means también de VLFEAT, pero de esta misma librería se complementa la asignación de las características a cada centroide con otra función llamada Kdtree la cual determina y relaciona cada centro calculado desde K-Means con las características almacenadas tras la aplicación de la función *vl_sift( )* .

Una vez la imagen es sometida a extracción de puntos de interés, cálculo de características respecto a esos puntos, agrupación de características y creación de relaciones clúster - características se genera un vector codificado llamado VLAD que tras pasar por una máquina de soporte vectorial entrega como resultados la exactitud general del clasificador sobre cada base de imágenes y los resultados de exactitud para cada clase que compone una base de imágenes.

Capitulo 4. Resultados del proyecto

En este capitulo se presentaran los resultados obtenidos una vez fue aplicada la metodología anterior. Debido a que se busca observar las diferencias y relaciones entre los resultados obtenidos a través de la clasificación por medio de redes neuronales convolucionales y los dos métodos presentados a partir de descriptores locales de características.

4.1 Clasificación de imágenes a través de la red neuronal convolucional VGG16

El primer caso de estudio fue la implementación de un clasificador de imágenes usando la red VGG16 entrenada previamente en Imagenet a través de las 10 bases de imágenes seleccionadas. Los resultados fueron considerados como el caso base del proyecto y a partir de este se realizaron las variaciones que conforman los casos experimentados con este tipo de clasificadores por CNN. La siguiente estructura define en forma simplificada la arquitectura y flujo de trabajo de la red.

Figura 4.1 Diagrama simplificado del modelo de red neuronal VGG16.

De igual forma que en la figura 4.1, se representa gráficamente la estructura de la red cuando se aplica el procedimiento de congelación de capas convolucionales buscando mejoras en los resultados. Siguiendo la lógica de la gráfica anterior, el cambio de color en el espacio “Block conv5” representa el bloque convolucional que será nuevamente entrenado a partir de las imágenes propias y en conjunto con los pesos pre entrenados de los anteriores 4 bloques de capas convolucionales.

Figura 4.2 Diagrama simplificado del modelo de red neuronal VGG16 para capas descongeladas

Tras un proceso de entrenamiento a través de 100 épocas, a una tasa de aprendizaje constante de 1E-3 y manteniendo la base de pesos pre-entrenados en los bloques convolucionales marcados con

color azul en conjunto con la etapa de clasificación del algoritmo en color verde en la figura 4.1 se obtuvo la siguiente tabla de resultados:

	Exactitud : VGG16 pre-entrenada en Imagenet				
Base de imágenes	VGG16 Caso base	Reentrenamiento: Capas Conv5.3+Conv5.2	Reentrenamiento: Capas Conv5.3	Reentrenamiento: Capas Conv5.2	Variación de exactitud
JUNIO20V1	77	67	76	77	[67 : 77]
FEBR20V3	65	60	60	58	[58 : 65]
JUNIO2OV3.	62	63	55	58	[55 : 63]
FEBR20V2	96	86	87	88	[86 : 96]
FEBR19V3.	93	97	96	98	[93 : 98]
FEBR19V2.	90	97	99	96	[90 : 99]
FEBR19V3IMAG18.	95	96	95	89	[89 : 96]
15-SceNe	87	77	84	82	[77 : 87]
Sports.	85	78	86	74	[74 : 86]
JUNIO20V2 (8 escenas)	78	71	75	65	[65 : 78]

Tabla 3Resultados de exactitud obtenidos para el clasificador usando VGG16-Imagenet

Nota: En la tabla 3 Las abreviaciones Conv5.3, Conv5.2 o la operación Conv5.3+Conv5.2 únicamente indican el nombre que por medio de las librerías Tensorflow y Keras se les asigna a las capas de cada bloque convolucional. El nombre en la columna indica que para estos resultados la capa fue descongelada y posteriormente reentrenada.

En la tabla 3 se puede observar el rendimiento de la red neuronal implementada como caso base(columna 1) y los resultados productos de descongelar ciertas capas convolucionales para ser reinicializadas y entrenadas nuevamente (columnas 3, 4 y 5). La última columna llamada variación de exactitud presenta los valores mínimos y máximos de clasificación tras aplicarse el proceso de descongelación de capas mencionado en el capítulo 3: caso 1.3.

De esta columna final se pudo obtener que un 50 por ciento de las bases de imágenes, es decir, 5 bases de imágenes : JUNIO20V3, FEBR19V3, FEBR19V2, FEBR19V3IMAG18, Sports. Estas obtuvieron mejoras entre un 1 y 7 %. Finalmente, en la siguiente grafica se presentan los datos obtenidos en la tabla 3.

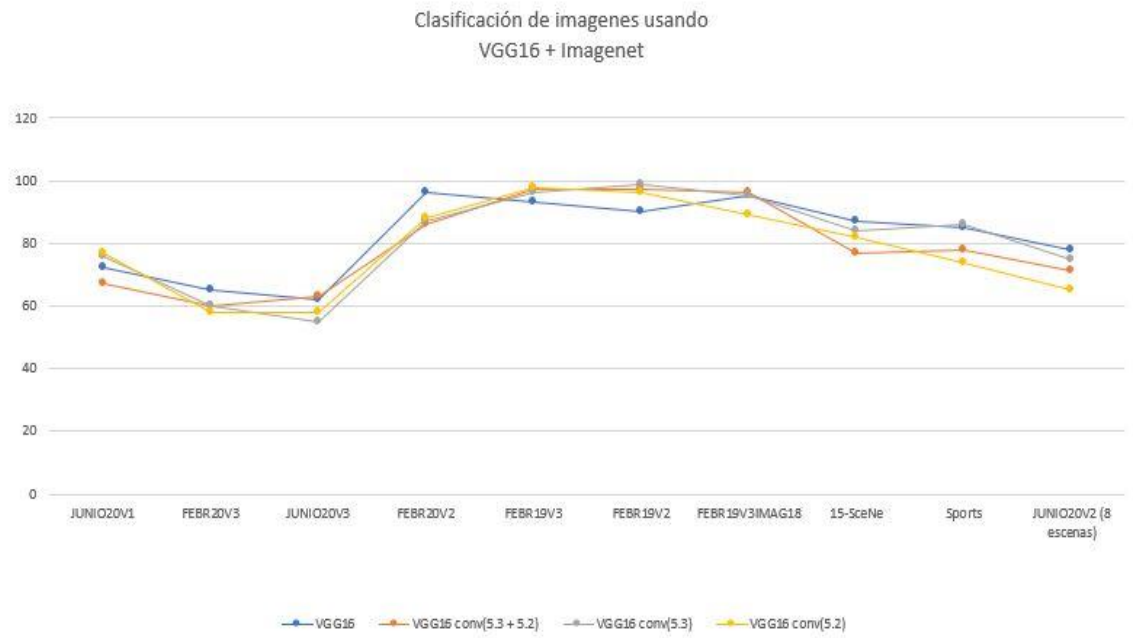

Figura 4.3 Grafica de exactitud contra modificaciones al caso base para cada una de las 10 bases de imágenes.

4.2 Clasificacion de imágenes a traves de la red neuronal VGG16 basada en Places365

Tal como se explica en el caso 1.2, el procedimiento de entrenar un clasificador de imágenes basado en pesos previamente calculados alrededor de Places365-Standard es exactamente igual al clasificador basado en Imagenet. Los resultados que se muestran a continuacion fueron calculados con dos excepciones respecto a los resultados encontrados en el numeral 4.1, una menor tasa de aprendizaje y un ligero cambio en el codigo para importar desde GitHub el nuevo conjunto de parametros para la red neuronal.

	Exactitud : VGG16 pre-entrenada en Places365			
Base de imágenes	VGG16 Caso base	Reentrenamiento: Capas Conv5.3+Conv5.2	Reentrenamiento: Capas Conv5.3	Variación de exactitud
JUNIO20V1	52	44	46	[44 : 52]
FEBR20V3.	49	49	53	[49 : 53]
JUNIO2OV3.	40	49	47	[40 : 49]
FEBR20V2.	68	69	71	[68 : 71]
FEBR19V3.	79	87	85	[79 : 87]
FEBR19V2.	82	90	88	[82 : 90]
FEBR19V3IMAG18	85	82	83	[82 : 85]
15-SceNe.	82	83	81	[81 : 83]

	Exactitud : VGG16 pre-entrenada en Places365			
Base de imágenes	VGG16 Caso base	Reentrenamiento: Capas Conv5.3+Conv5.2	Reentrenamiento: Capas Conv5.3	Variación de exactitud
Sports	79.	77	82	[77 : 82]
JUNIO20V2	75.	66	77	[66 : 77]

Tabla 4Resultados de exactitud obtenidos para el clasificador usando VGG16-Imagenet

En la tabla 4 Se presentan resultados de clasificación para cada base de imágenes tanto para el caso base, es decir, actualizando los parámetros de las capas completamente conectadas y de clasificación como para el entrenamiento cuando se descongelan las capas conv5.3 y conv5.2. En esta ocasión debido a no obtener resultados confiables cuando la capa convolucional 5.2 se descongeló no fue posible registrarlos en la tabla. De la comparación entre las columnas 2, 3 y 4 y los intervalos registrados en la columna 5 se pudo determinar que bases de imágenes obtuvieron en los experimentos siguientes mejores resultados del modelo base.

Del análisis de las columnas mencionadas se pudo obtener que un 80 por ciento de las bases de imágenes, es decir, 8 bases de imágenes : FEBR20V3, JUNIO20V3, FEBR20V2, FEBR19V3, FEBR19V2, 15-SceNe,Sports, JUNIO20V2. Estas obtuvieron mejoras entre un 1 y 9 %. Finalmente, en la siguiente grafica se presentan los datos obtenidos en la tabla 4.

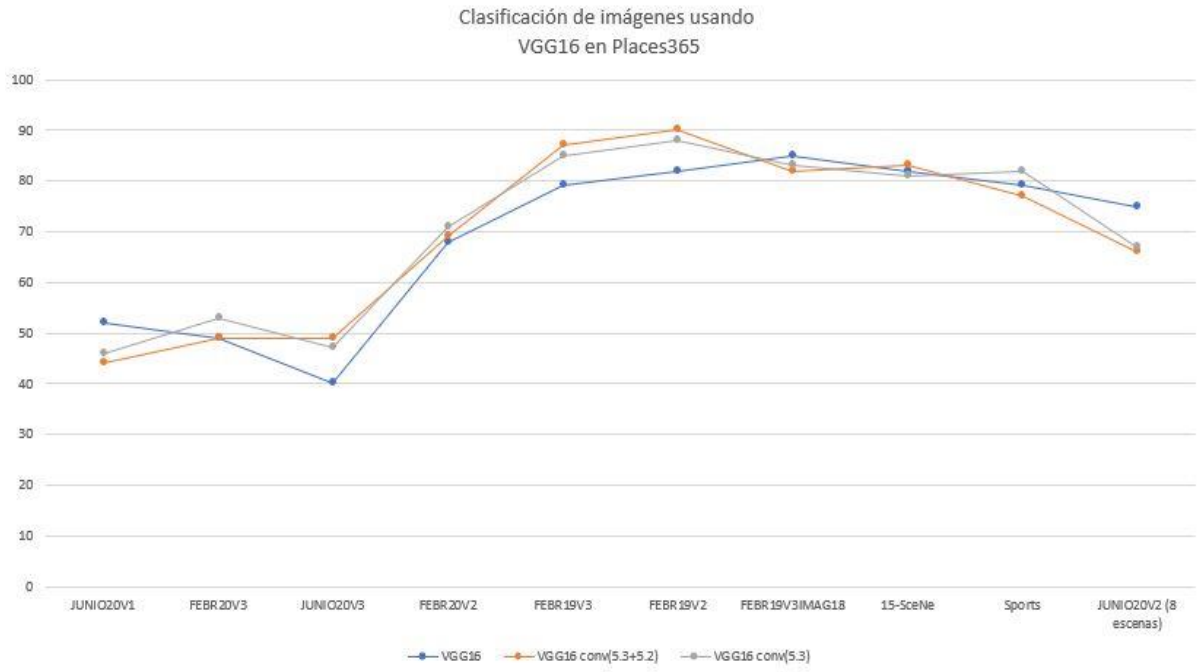

Figura 4.4 Grafica de exactitud contra modificaciones al caso base en Places365 para cada una de las 10 bases de imágenes.

Una vez obtenidos todos los resultados de clasificación sobre el modelo de red neuronal convolucional VGG16 y al no encontrar un patron claro entre las modificaciones aplicadas a este y los resultados positivos en cada base de imágenes, se decide comparar los resultados del modelo ilustrado en la figura 1 tanto en Places365 como base como en Imagenet. Estos son los dos modelos principales en cada caso. La siguiente figura muestra graficamente lo aquí escrito.

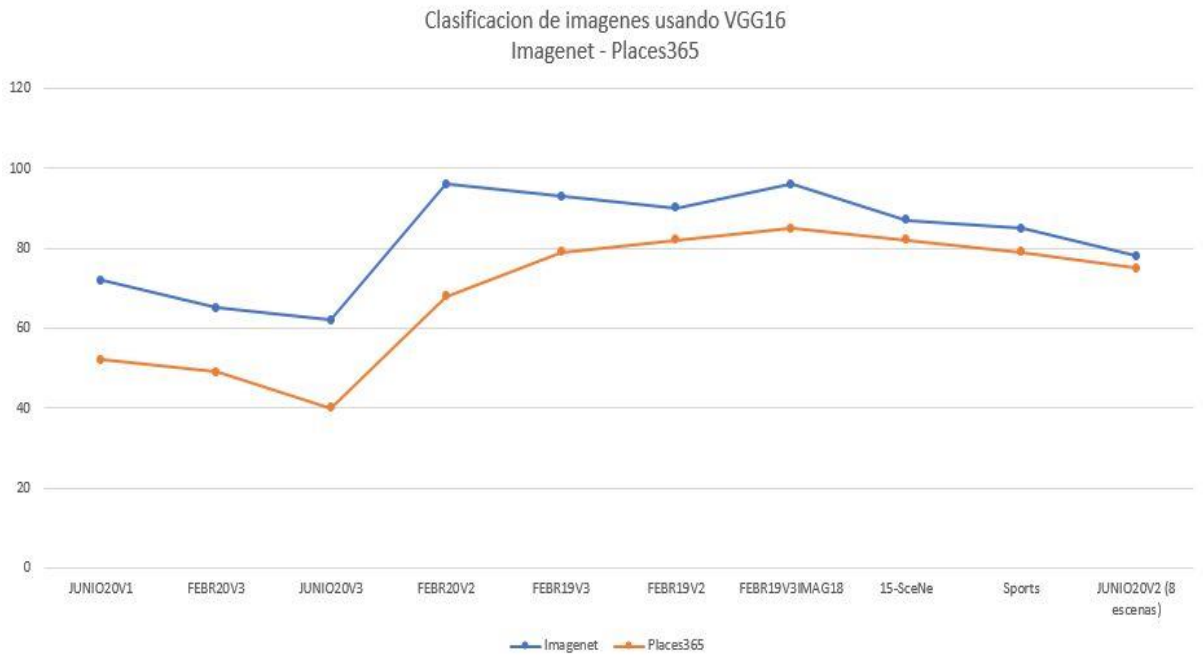

Figura 4.5 Grafico comparativo entre los resultados de clasificacion del modelo VGG16 pre-entrenado tanto en Places365 como en Imagenet sin ninguna mejora o modificacion aplicada a su estructura.

4.3 Clasificación de imágenes a través de la red neuronal convolucional Resnet50(Residual Network)

Para el caso del modelo de red neuronal Resnet50 implementado en Pytorch bajo las mismas 8 bases de imágenes usadas anteriormente, se omite la etapa de entrenamiento bajo los pesos calculados previamente en Imagenet y se llega directamente al modelo de Resnet50 previamente capacitada en Places365. Debido a que fue posible recrear los mismos procesos de : adaptación de la capa de clasificación softmax a 5, 8 y 15 categorias respectivamente e igualmente que en anteriores procedimientos descongelar porciones del modelo para ser reinicializadas y entrenadas nuevamente, se obtuvo una tabla con la exactitud general del modelo para cada una de las posibles combinaciones y actualizaciones posibles en el bloque convolucional final del modelo que explorado desde la librería Pytorch se nombraban sus capas como conv4.1, conv4.2 y conv4.3. La siguiente tabla presenta los resultados en general del modelo de clasificación.

	Exactitud : Resnet50 pre-entrenada en Places365				
Base de imágenes	Resnet50 Caso base	Reentrenamiento: Capas Conv4.3+Conv4.2	Reentrenamiento: Capas Conv4.3	Reentrenamiento: Capas Conv4.2	Variación de exactitud
JUNIO20V1	84	88	89	84	[84 : 89]
FEBR20V3	93	96	92	92	[92 : 96]
JUNIO2OV3.	89	93	91	91	[89 : 93]
FEBR20V2	100	100	100	100	[100]
FEBR19V3.	100	99	100	100	[99 : 100]
FEBR19V2.	100	100	100	100	[100]
FEBR19V3IMAG18.	100	96	100	100	[96 : 100]
15-SceNe	94	94	96	96	[94 : 96]
Sports.	96	96	94	94	[94 : 96]
JUNIO20V2 (8 escenas)	90	91	90	90	[90 : 91]

Tabla 5Resultados de exactitud obtenidos para clasificación de imágenes usando Resnet50-Places365

Para este proceso de clasificación basado en Resnet50 se obtuvieron valores tanto inferiores y superiores como iguales en la mayoría de las bases de imágenes, es decir no se presenta un patrón en las distintas pruebas que indique el efecto de inicializar los pesos de las capas convolucionales para ser calculados nuevamente. Los valores de exactitud variaban según la base de imágenes a utilizar entre 84 y 100 %. Al menos 6 de las 10 bases de imágenes presentaron en alguna prueba un valor superior al del caso base. Estas son : JUNIO20V1, FEBR20V3, JUNIO20V3, 15-SceNe, Sports, JUNIO20V2.

La siguiente grafica presenta los resultados obtenidos en la tabla 4.3, permitiendo observar que tanto varían los resultados en cada base de imágenes en funcion del tipo de pruebas realizadas.

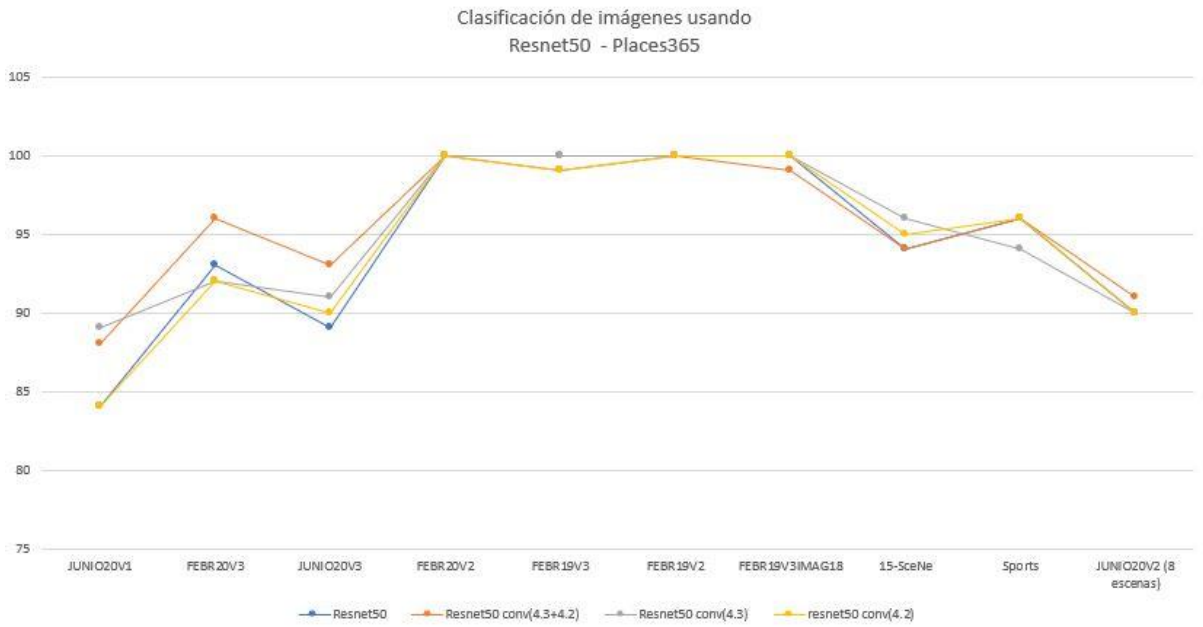

Figura 4.6 Grafico comparativo del clasificador de imágenes basado en Resnet50 frente a las variantes propuestas y realizadas sobre los parámetros de algunas de sus capas convolucionales finales.

4.4 Clasificación de imágenes usando BOVW

El clasificador basado en bolsa de palabras visuales fue implementado en Matlab sobre un número reducido de imágenes en comparación con el clasificador basado en CNN. Se evaluó su rendimiento sobre 25 imágenes por clase para cada una de las 6 bases de imágenes utilizadas obteniéndose los siguientes resultados.

Base de imágenes	Exactitud : BOVW (%)
JUNIO20V1	84.80
FEBR20V3	64.00
JUNIO20V3	37.60
FEBR20V2	100.00
15-SceNe	60.00
Sport	77.50

Tabla 6Resultados de clasificación usando descriptor local de características BOVW

De igual forma se obtuvieron los vectores de clasificación por categoría para cada base :

Base de imagenes	Probabilidad por clase (%) C1 C2 C3 C4 C5	Exactitud del clasificador (%) BOVW
JUNIO20V3	[68 24 48 0 48]	37.60
FEBR20V3	[68 56 72 80 48]	64.00
JUNIO20V1	[100 100 96 28 100]	84.80
FEBR20V2	[100 100 100 100 100]	100

Tabla 7Resultados de clasificación por categoría usando BOVW – parte I

Para los conjuntos de imagenes SceNe-15 y Sport se generará una nueva tabla debido a que no comparten las mismas categorias de las imágenes presentadas en la tabla 6. A continuación se presentan los resultados para los dos conjuntos de imágenes restantes.

Base de imágenes	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	Exactitud (%)
15-SceNe	24	88	40	76	76	88	68	88	52	36	28	92	100	20	28	60.27
	Badm		Bocce		Croquet		Polo		RockClimb		Rowing		Sailing		Snowboarding	
Sport	96		36		72		76		92		64		92		92	77.50

Tabla 8Resultados de clasificación por categoría para las bases de imágenes 15-SceNe y Sport.

Finalmente, en el caso del clasificador usando BOVW como descriptor se presentan de forma grafica los resultados obtenidos en la tabla 6.

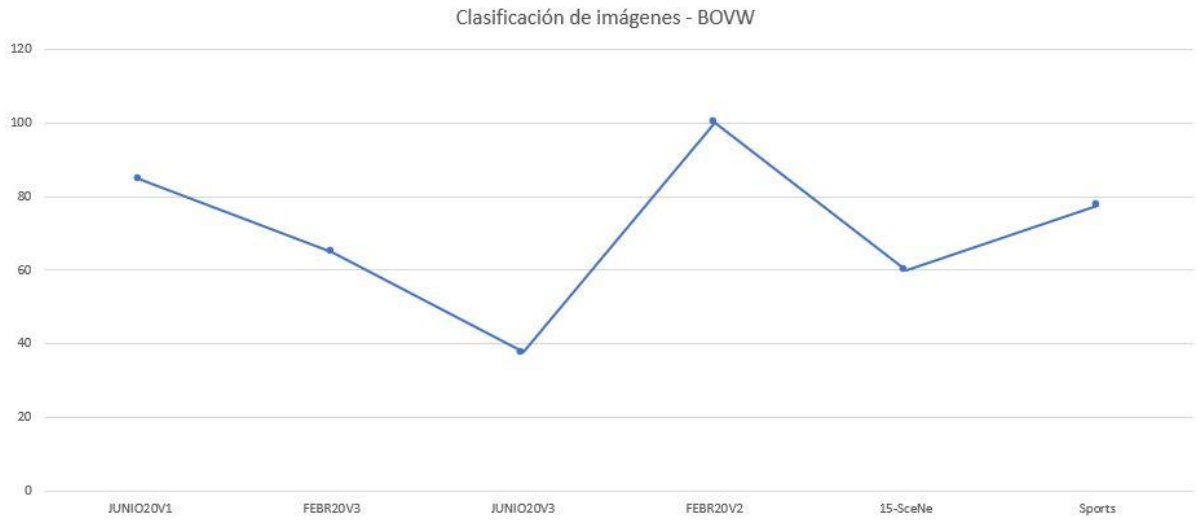

Figura 4.7 Representación gráfica de los resultados registrados para el clasificador de imágenes usando BOVW.

Se puede observar que comparte un comportamiento similar entre una base de imágenes y otra, es decir la relación entre qué base de imágenes tiene mejor resultado respecto a otra se conserva parcialmente en comparación con la clasificación de imágenes usando como CNN a Resnet50, aunque los valores como tal no sean los mismos.

4.4 Clasificación de imágenes usando VLAD

Al igual que BOVW el descriptor de características locales VLAD también fue implementado en Matlab, aplicándose solo a las primera cuatro bases de imágenes usadas por el clasificador de imágenes

basado en BOVW. El número de clústeres con el cual se realizaron las pruebas de clasificación fue 64.Por medio de una máquina de soporte vectorial y el vector codificado de las imágenes se obtuvieron los siguientes resultados :

Base de imágenes	Exactitud : BOVW (%)
JUNIO20V1	78.40
FEBR20V3	71.20
JUNIO20V3	37.60
FEBR20V2	99.20

Tabla 9 Resultados de clasificación usando descriptor local de características VLAD

Una vez registrados los resultados en la tabla anterior, se explora por medio de la siguiente tabla los resultados de clasificacion por clase para cada uno de los anteriores conjuntos de imágenes con el fin de determinar que clases tuvieron un mejor rendimiento.

Base de imagenes	Probabilidad por clase (%) C1 C2 C3 C4 C5	Exactitud del clasificador (%) BOVW
JUNIO20V3	[28 80 40 52 60]	37.60
FEBR20V3	[60 56 76 76 88]	71.20
JUNIO20V1	[88 92 100 24 88]	78.40
FEBR20V2	[100 96 100 100 100]	99.20

Tabla 10Resultados de clasificación por categoría usando descriptor local de características VLAD

Obtenidos los resultados se presenta la gráfica extraída de los datos de clasificación en la tabla 4.7.

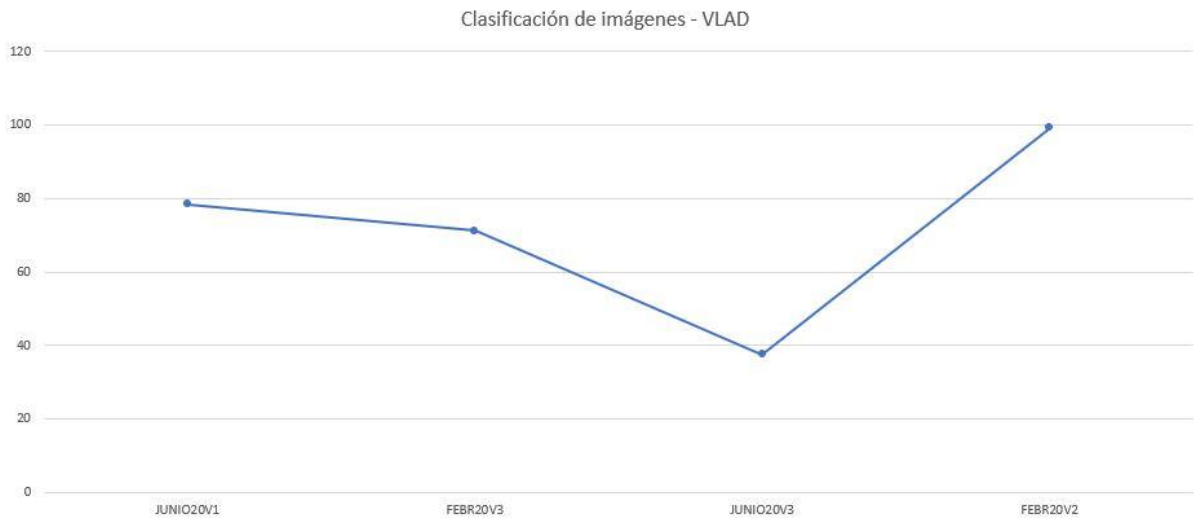

Figura 4.8 Representación gráfica de los resultados registrados para el clasificador de imágenes usando VLAD.

Al igual que en todos los modelos de clasificación de imágenes presentados en este documento JUNIO20V3 sigue siendo el conjunto de imágenes con uno de los peores resultados en general. Logrando entre 46 y 75 imágenes acertadas de un total de 125 imágenes, solo superando estos límites

al evaluarse en Resnet50 donde alcanzó valores de exactitud en general entre 89 y 93 por ciento, es decir logrando clasificar correctamente entre 111 y 116 imágenes. Finalmente, para cerrar la etapa de presentación de resultados se presenta una gráfica comparativa entre los dos métodos basados en descriptores de características locales VLAD y BOVW.

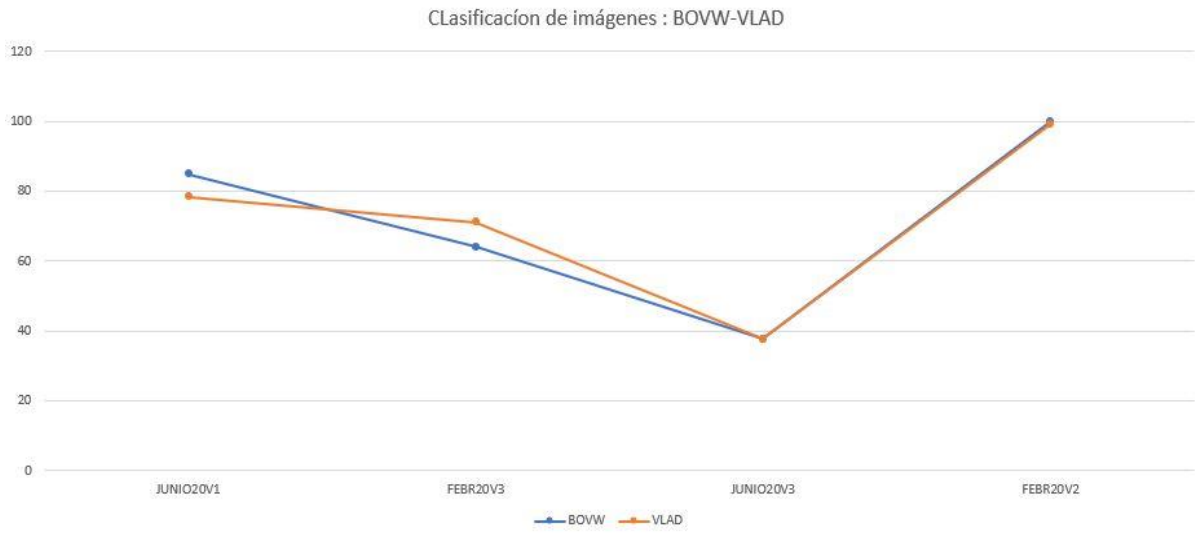

Figura 4.9 Grafica comparativa entre los dos clasificadores basados en descriptores locales de características.

Como ultima comparación y teniendo en cuanto que según los resultados registrados en esta sección los mejores resultados se obtuvieron bajo los considerados casos base de cada arquitectura, es decir, los casos sobre los cuales se inició a realizar distintas pruebas hablando específicamente de VGG16 previamente entrenada en Imagenet, Resnet50 usando como parámetros los importados desde el sitio web de Places365 y las dos versiones de descriptores locales BOVW y VLAD en las únicas 4 imágenes que compartían los cuatro modelos.

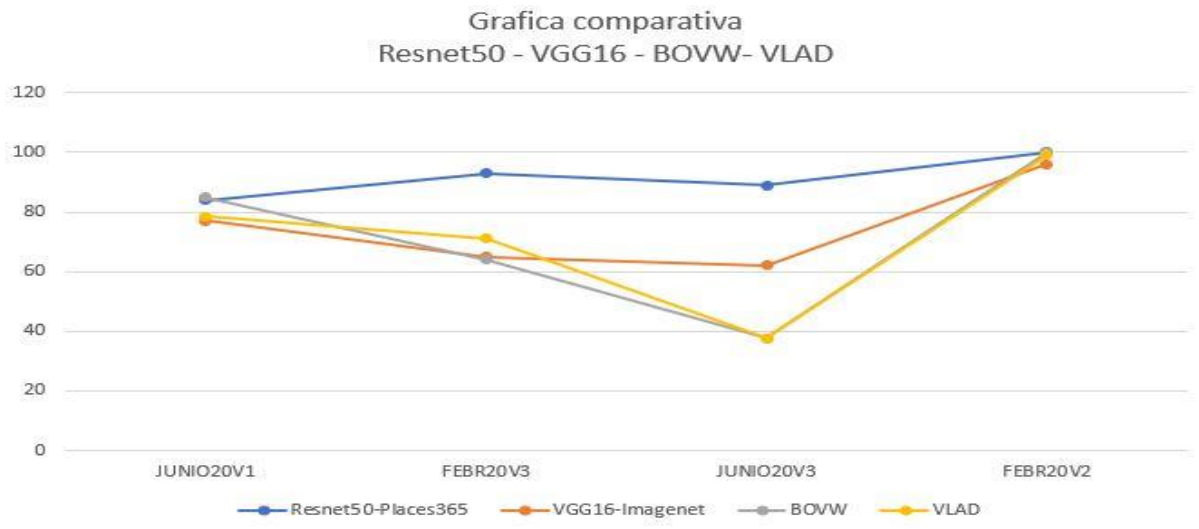

Figura 4.10 Grafica comparativa basada en el caso base implementado de cada modelo de clasificación.

En la figura 4.10 puede observarse la superioridad de sus resultados en los cuatro conjuntos de imágenes evaluados y como los dos metodos de clasificacion basados een descriptores locales terminaron teniendo resultados similares.

Capítulo 5. Conclusiones

- Como resultado final de este proyecto se obtuvo un algoritmo que permite clasificar conjuntos de imágenes con distintas características. Desde escenas completas hasta porciones de ellas fue posible discriminar correctamente. Además, se logró evidenciar de forma gráfica como distintos tipos de redes neuronales convolucionales pueden conservar guardando las proporciones una respuesta similar ante condiciones similares en el proceso de capacitación del algoritmo.

- También se evidenció el potencial existente en redes neuronales que previamente han sido capacitadas con imágenes similares a las propias, pero en mayor dimensión, el utilizar una base de parámetros pre-entrenados que fueron ajustados tras el procesamiento de más de 1 millón de imágenes hizo que la carga computacional en las pruebas con las bases de imágenes propias fuera menor y el tiempo de ejecución se redujera al tiempo que mejoraban los resultados.

- En cuanto al efecto de la profundidad de la red, es decir, el número de capas convolucionales y de agrupación de VGG16 en relación con los resultados es notable. Al no poder extraer características hasta cierto nivel su rendimiento fue siempre inferior.

- La composición de las imágenes también afectó el rendimiento de ambas pruebas tanto con redes neuronales convolucionales como con descriptores locales de características, esto debido a que en el momento de validar el funcionamiento responden con mejores resultados a escenas como objeto de prueba que a objetos o partes de ellos.

- Finalmente, en cuanto a los descriptores locales de características se refiere, tanto VLAD como Fisher tuvieron una respuesta inferior al clasificador realizado alrededor de resnet50, pero otra característica especial es su respuesta negativa a la base de Imágenes JUNIO20V3 el cual se evidencia incluso en la clasificación por redes neuronales. Aun así, cabe resaltar que tanto VLAD como BOVW con resultado muy cercanos con una diferencia máxima entre uno y otro de aproximadamente 7% en el caso más extremo obtuvieron un mínimo de 3 categorias superiores al 70% de rendimiento teniendo en cuenta que solo contaban con un máximo de 100 imágenes por clase y no existía el efecto de las características aprendidas anteriormente.

Capítulo 6. Recomendaciones

- La forma en la cual se registran los datos podría mejorar. Para este caso solo se tuvo en cuenta el porcentaje de acierto de cada modelo y la precisión en cada clase, pero dado que aun tomando varias medidas los resultados pueden variar ligeramente tras cada etapa de entrenamiento sería una buena práctica evaluar en que clases hay mayor variación y que tan grande es el margen de error en cada caso que se repite.

- Otra practica que podría ayudar a mejorar los resultados seria variar el tipo de funcion que extrae los puntos de interés, dado que cada método de detección de puntos de interés y extracción de caracteristicas evalúa y registra diversos tipos de detalles. Algunos podrían ser menos susceptibles al error ante imágenes recortadas o ampliadas, de la misma forma que se explora diversos mecanismos de extracción de caracteristicas es posible variar ciertas condiciones del proceso de clustering como el tamaño del Clúster, a ensayo y error se podría encontrar un modelo que se ajuste a esas bases que tanto en CNN como en VLAD o BOVW presentan los resultados más negativos.

Capitulo 7. Bibliografia

[1] very high resolution image scene classification with semantic Fisher Vectors. Souleyman Chaiba.c, Yanfeng Gub. Hongxun Yaoa. Khaled Belkadic School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China Department of Information Engineering, Harbin Institute of Technology, Harbin, China School of Computer Science and Technology, University of Science and Technology, Oran, Algeria[2018)

[2] B. Zhao, Y. Zhong, and L. Zhang, "A spectral structural bag-offeatures scene classifier for very high spatial resolution remote sensing imagery," {ISPRS} Journal of Photogrammetry and Remote
Sensing, vol. 116, pp. 73 – 85, 2016.

[3] J. Zou, W. Li, C. Chen, and Q. Du, "Scene classification using local and global features with collaborative representation fusion," Inf. Sci., vol. 348, pp. 209–226, 2016.

[4] Xiaojuan Cheng; Jiwen Lu; Jianjiang Feng; Bo Yuan; Jie Zhou. "Scene recognition with objectness". ScienceDirect Pattern recognition 2017.

[5] Mandar Dixit;Si Chen; Dashan Gao; Nikhil Rasiwasia;Nuno Vasconcelos." Scene Classification with Semantic Fisher Vectors". IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015.

[6] Mandar, Dixit. Yunsheng, Li.Vasconcelos,Nuno. Semantic Fisher Scores for Task Transfer:
Using Objects to Classify Scenes. JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015

[7] C.Y. Lee; S. Xie; P. Gallagher. "Deeply-supervised nets". AISTATS, 2015.

[8] Mandar, Dixit. Yunsheng, Li.Vasconcelos,Nuno. Semantic Fisher Scores for Task Transfer:
Using Objects to Classify Scenes. JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015

[9] B. Zhou, A. Lapedriza, J. Xiao, Learning deep features for scene recognition using places database, in: NIPS, 2014, pp. 487–495

[10] L.-J. Li;H. Su; Y. Lim; and F.-F. Li." Object bank: An object-level image representation for high-level visual recognition". International Journal of Computer Vision, 2014.

[11] W. Shao, W. Yang, G. Xia, and G. Liu, "A hierarchical scheme of multiple feature fusion for high-resolution satellite scene categorization," in Computer Vision Systems - 9th International Conference, ICVS 2013, St. Petersburg, Russia, July 16-18, 2013. Proceedings, 2013, pp. 324–333.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[13] R. Kwitt, N. Vasconcelos, and N. Rasiwasia. Scene recognition on the semantic manifold. In Proceedings of the 12th European conference on Computer Vision - Volume Part IV, ECCV'12, pages 359–372, Berlin, Heidelberg, 2012. Springer-Verlag.

[14] C. Li, D. Parikh, T. Chen, Automatic discovery of groups of objects for scene understanding, in: CVPR, 2012, pp. 2735–2742.

[15] M.J. Choi, J.J. Lim, A. Torralba, Exploiting hierarchical context on a large database of object categories, in: CVPR, 2010, pp. 129–136.

[16] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, JMLR 3 (1) (2003) 993–1022.

[17] S. Geman, C. Graffigne, Markov random field image models and their applications to computer vision, in: ICM, 1986, p. 2.

[18] Fenoll, Fran; Ramirez, Fran; Blanco, Enrique.CDCO Telefonica. (4 de junio, 2020). Las matemáticas del Machine Learning: Funciones de activación. https://empresas.blogthinkbig.com/las-matematicasdel-machine-learning-funciones-de-activacion/. Articulo versión web.

[19]Jordi Torres i Viñals.(2018).Deep Learning- Introducción practica con Keras.Pdf. Editorial Marcombo. Disponible en : https://torres.ai/python-deep-learning/.

[20] OpenCV 4.5.2-pre. (Actualizada a 18 de marzo 2021). Open Source Computer Vision.Feature detection and description. Disponible en: https://docs.opencv.org/master/db/d27/tutorial_py_table_of_contents_feature2d.html

[21] Müeller, Andreas C. y Guido, Sarah. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists. Editorial O'REILLY.

[22] Unpingco, José. (2016). Python for Probability, Statistics, and Machine Learning. Second Edition. Editorial Springer.

[23] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron. (2016). Deep learning. MIT Press.

[24] Géron, Aurélien. [2016]. Hands-On Machine Learning with Scikit-Learn and TensorFlow. Editorial O'REILLY

[25] Universitat Autónoma de Barcelona. Coursera. Clasificación de imágenes : ¿Cómo reconocer el contenido de una imagen?

[26] Documentación OpenCV- Maquinas de soporte vectorial. Material online. OpenCV Team. Disponible en : https://docs.opencv.org/3.4/d1/d73/tutorial_introduction_to_svm.html

[27] Clasificación de imágenes : ¿Cómo reconocer el contenido de una imagen? Coursera. Universitat Autónoma de Barcelona. Disponible en : Universitat Autónoma de Barcelona. Disponible en: https://www.coursera.org/learn/clasificacion-imagenes

[28] ¿Qué es Colaboratory ?. Google. definición e introducción. Disponible en :https://colab.research.google.com/notebooks/intro.ipynb

[29]Documentación : Segmentación semántica. Matlab & Simulink. (2021- actualizado). Disponible en: https://la.mathworks.com/solutions/image-video-processing/semantic-segmentation.html

[30]Bryan C. Russell*, Antonio Torralba*.Kevin P. Murphy. William T. Freeman.LabelMe: a database and web-based tool for image Annotation.2008. Disponible en https://people.csail.mit.edu/brussell/research/AIM-2005-025-new.pdf

[31]Docs : Annotate data with labelme . Dlology.Disponible en:https://www.dlology.com/blog/how-to-create-custom-coco-data-set-for-instance-segmentation/ . Version online

[32]Keras. Keras: Transfer learning & Fine Tunning. Disponible en: https://keras.io/guides/transfer_learning/ . Version online.

[33]Docs :IBM Cloud Annotations. Disponible en : https://cloud.annotations.ai/docs

[34]Pesos pre-entrenados VGG16-Places365. Disponible en : https://github.com/CSAILVision/places365

[35] Definición sobre librería VLFeat. VLFeat About. Disponible en :https://www.vlfeat.org/about.html

[36]Pytorch : documentation. Pytorch 1.8.1. Disponible en: https://pytorch.org/docs/stable/index.html